ARMY RESEARCH LABORATORY

Implementation of the Nonlinear Composite Analysis Code "LAMPAT" Into LLNL-DYNA3D

by Ala Tabiei and George A. Gazonas

ARL-TR-2846 September 2002

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5069

ARL-TR-2846 **September 2002**

Implementation of the Nonlinear Composite Analysis Code "LAMPAT" Into LLNL-DYNA3D

Ala Tabiei
University of Cincinnati

George A. Gazonas
Weapons and Materials Research Directorate, ARL

Abstract

The nonlinear composite analysis code "LAMPAT" is implemented in the nonlinear explicit finite element code LLNL-DYNA3D as a new user-defined material model. All subroutines of LAMPAT are implemented as user-defined Material Model 46. The user-defined material subroutine calls an external data base file that contains material properties for several composites. In addition, LAMPAT is modified for use in an explicit time-integration solver. The model is improved to account for loss of symmetry of the material stiffness matrix resulting from degradation of the elastic moduli during damage evolution. The model implementation is validated through a one-element simulation and penetration simulations. In addition, comparing the prediction of the LSDYNA elastic material model with that of LAMPAT validates the implementation.

Acknowledgments

I would like to thank Dr. George Gazonas for inviting me to work at the U.S. Army Research Laboratory (ARL). I appreciate the guidance and help he provided me. I would like to thank ARL for providing me with this valuable opportunity to visit and interact with in-house scientists and researchers. The financial support provided by the Oak Ridge Institute for Science and Education program is acknowledged. I would also like to thank Dr. Bruce Fink, ARL, for providing support to do my work successfully during this period.

INTENTIONALLY LEFT BLANK.

Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The U.S. Army Research Laboratory (ARL) is actively seeking better computational material models to assist the design and optimization process of armors. The composite lightweight integral armor under consideration contains a variety of materials, such as fiber-reinforced composites, metals, loss fabric, ceramics, and rubber. To be able to simulate penetration and perforation, a robust computer code must be utilized. The nonlinear explicit finite element code LLNL-DYNA3D* [1] is an excellent candidate to successfully simulate such problems. However, the code is very general and needs to be augmented with user-defined material models. The user-defined material model options in DYNA3D allow users to customize the code with advanced material models of interest to the armor community. Researchers at ARL have been developing the "LAMPAT" nonlinear composite analysis code for the analysis of thick-section composite structures for more than 10 years [2]. LAMPAT models the material nonlinear behavior observed in some composites and is based on homogenization procedures. To be able to design armored vehicles, one must be able to simulate the high-speed ballistic impact event with good accuracy. In this manner, one can use numerical simulation to optimize an armor that is able to contain high-speed projectiles. The material model developed in-house at ARL can perform this task provided that it is implemented in a wave propagation code like DYNA3D. The LAMPAT code was implemented in an initial effort [3] into the commercial code LSDYNA [4] as a user-defined material subroutine. However, the source code for the commercial code is not available for further development, which makes it more difficult to link to more sophisticated armor optimization algorithms [5].

The implementation of the LAMPAT code into the DYNA3D code herein is described. The modification of the nonlinear LAMPAT code is necessary for successful utilization of the code in the finite element method. This modification is described next. The implementation is validated through several examples that are described and presented in the following sections. A listing of the modified LAMPAT code is presented in Appendix A. To understand the solid element formulation, Appendix B lists the algorithm that completely describes this element.

2. Implementation

The original LAMPAT development is performed for analysis of composite structures with a mechanics of composite materials approach. In that approach, the stresses are incrementally

* LLNL-DYNA3D is the Lawrence Livermore National Laboratories version of the finite element computer code hereafter referred to as simply DYNA3D.

increased, and the corresponding strains are obtained (or vice versa). The original formulation leads to numerical instability due to loss of symmetry of the material stiffness matrix when implemented into three-dimensional finite element codes like DYNA3D. The loss of symmetry in the material stiffness matrix is attributed to how the elastic constants are degraded. The material reciprocity relation for an orthotropic material is given by

$$\frac{\nu_{ij}}{E_i}=\frac{\nu_{ji}}{E_j} \quad , \quad i,j=1,2,3. \tag{1}$$

The material compliance matrix for an orthotropic material is

$$[C]=[S]^{-1}=\begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{12}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix}^{-1} \tag{2}$$

The compliance matrix can lose symmetry if material constants are arbitrarily degraded during damage progression. The loss of symmetry is avoided by utilizing the following compliance matrix:

$$[C]_y=[S]_y^{-1}=\begin{bmatrix} \frac{1}{E_1} & -\sqrt{\frac{\nu_{12}}{E_1}\frac{\nu_{21}}{E_2}} & -\sqrt{\frac{\nu_{12}}{E_1}\frac{\nu_{21}}{E_2}} & 0 & 0 & 0 \\ -\sqrt{\frac{\nu_{12}}{E_1}\frac{\nu_{21}}{E_2}} & \frac{1}{E_2} & -\frac{\nu_{23}}{\sqrt{E_2E_2}} & 0 & 0 & 0 \\ -\sqrt{\frac{\nu_{12}}{E_1}\frac{\nu_{21}}{E_2}} & -\frac{\nu_{23}}{\sqrt{E_2E_2}} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix}^{-1} \tag{3}$$

The material nonlinearity treatment in explicit finite elements is different than that for implicit finite elements. In implicit finite elements, we have load increments and equilibrium iterations. The tangential stiffness, due to material nonlinearity, must be updated at each load increment and should also be updated at each equilibrium iteration. The material nonlinear behavior in LAMPAT is summarized by the following equation:

$$\Delta\sigma_{ij} = \frac{E_{ijo}}{\left(1+\left(\frac{E_{ijo}\varepsilon_{ij}}{\sigma_{oij}}\right)^{n}\right)^{1+\frac{1}{n}}}\Delta\varepsilon_{ij}. \tag{4}$$

The concept of equilibrium is given by equation 5. When the internal force vector is equal to the external force vector, equilibrium is satisfied. The internal force vector is the integral over the volume of the stresses times the strain displacement matrix of the finite element. The internal force vector is defined by equation 6.

$$R^{\text{int}} = R^{ext}. \tag{5}$$

$$R^{\text{int}} = \int B^{T}\sigma_{ij}dv. \tag{6}$$

In implicit finite elements, the external load is incremented in several steps. At the first iteration of the first step, the internal forces are normally not in equilibrium with the external forces. The out-of-balance force is called the residual ΔR. Due to this residual, a correction to the displacement is obtained by inverting K in equation 7 and solving for the displacement.

$$R^{\text{int}} - R^{ext} = \Delta R = [K]\{\Delta u\}. \tag{7}$$

$$\Rightarrow \Delta u \Rightarrow \Delta\varepsilon \Rightarrow \Delta\sigma. \tag{8}$$

From the corrected displacement, a new strain increment is obtained, and a new stress increment can be calculated (equation 8). Subsequently, new internal forces are obtained. These iterations are continued until the out-of-balance force is less than a user-defined value.

The iteration algorithm previously described can be understood by examining Figures 1 and 2. At the start of the load step "n" there is a tangential material stiffness update. As equilibrium iterations proceed, the material stiffness matrix is also updated as depicted in Figure 2.

In explicit finite elements, however, the load varies incrementally at each time step. There is no equilibrium iteration, since the acceleration is solved for in the same time step. The acceleration is integrated once to obtain velocity and once again to obtain displacement. In explicit finite elements, we have many load increments due to many cycles (time steps) to maintain time integration stability that is governed by the Courant condition. The integration time steps can be on the order of microseconds or fractions of microseconds. Consequently, in a typical impact simulation thousands of cycles are possible. This fact can render the frequent update of the tangential material stiffness matrix computationally inefficient. LAMPAT calculations are computationally costly due to several transformations of stresses and strains for multiple layers from global to local and back from local to global frames of reference. In explicit finite elements, the LAMPAT material stiffness matrix updates must be limited to achieve computational efficiency for large-scale finite element impact simulations. For this reason, the tangential material stiffness matrix is stored in history variables for subsequent use in the calculation. The current implementation permits the user to define the frequency of the material stiffness matrix update.

The initial verification of the implementation is performed on an isotropic elastic material. A linear elastic material behavior is obtained in LAMPAT by suppressing the power law stress-strain relation (equation 4). The material model is then tested for prediction of stress for an isotropic elastic material by keeping the composite material transformation matrices intact. Exact prediction is obtained as compared to the existing material elastic material model in DYNA3D. Finally, the model is tested for material nonlinear behavior.

3. Results

A finite element model of a block consisting of one element is developed for verification of the nonlinear behavior. The block is constrained at one end and a prescribed displacement is applied to the other end. Two material systems are considered for validation of the implementation. An isotropic elastic material (aluminum) and nonlinear elastic composite (graphite/epoxy) are considered. The linear elastic material is not shown here, as it leads to the same behavior of the original elastic material model in DYNA3D. Figures 3–5 depict the stress vs. strain for the material system considered. The validation of the nonlinear behavior is carried out in the three coordinate directions (i.e., x-, y-, and z-directions). The nonlinear material behavior is illustrated in these plots.

Next, the implementation is tested for numerical stability and the ability to simulate an impact event. For this purpose, two models are considered. The first model is a ball impacting thick plates (Figures 6 and 7). The plates are made of graphite/epoxy with nonlinear material behavior. The impact velocity is 1.5e+4 in/s. The second example considered is a flat plate impacting a curved plate. The flat plate is given two velocity components. One component (2.0e+4 in/s) is in the normal direction and the other velocity component (1.0e+4 in/s) is in the circumferential direction. The model and the result of the simulation are depicted in Figures 8 and 9. The flat plate is made of graphite/epoxy with nonlinear material behavior.

Finally, a 12 × 12 × 1.6-in composite plate made of S-2 Glass*/epoxy was constructed, clamped at four corners, and impacted with a .50-cal. (1.6-oz) trimetallic bullet consisting of a copper jacket, steel core, and lead-tipped filler. The experimentally determined V50 velocity was 17,280 in/s (1440 fps), although this value was not known when the simulations were performed. Three different composite material models were evaluated to see which could best predict the V50 that was determined experimentally. The first model is the orthotropic elastic Material Model 22 in LSDYNA. The second model is the micromechanical strain-rate-sensitive unidirectional composite material model described by Tabiei and Chen [6] and implemented as a user-defined material subroutine in LSDYNA. The third material model is the new LAMPAT/DYNA3D implementation. The prediction of each of the material models is considered and described in the next section.

1.1 Material 22 in LSDYNA

The composite plate is modeled with 45 layers of solid elements through the thickness. Each layer represents a layer of unidirectional composite. Two layers together, one in the zero direction and the other in the 90 direction, represent one layer of plain weave composite. The material (S-2 Glass/epoxy) properties for Material Type 22 used in the simulation are listed in Table 1. Several simulations were carried out with bullet velocities that ranged from 13,000 to 16,000 in/s. Figures 10–15 depict the results for varying the impact velocity to find the V50. Figure 10 shows the penetration of the bullet into the plate, the velocity of the bullet as a function of time, and the displacement of the center of the plate as a function of time. The velocity of node 52569 is shown in Figure 10. This node is on the center of the back of the bullet. The displacement of node 1 is also presented in Figure 10. This node is on the center back face of the plate. It can be seen that the bullet is stopped in the plate with an impact velocity of 13,000 in/s. Figures 11 and 12 depict results for impact velocities of 13,500 and 14,000 in/s, respectively. In both of these simulations, the bullet stopped at about 250 ms. The maximum deflection of node 1 in both cases is around 0.03 in. Note that for the case of the 14,000 in/s impact velocity, the maximum deflection is 0.14 in. This is due to the eroding algorithm. When an element fails, it is removed from the database. The nodes of the removed element are also removed. The sharp increase of the displacement in Figure 12 has no physical meaning. Figure 13 depicts the V50 velocity results (14,500 in/s). The impact velocity is increased to 15,000 and 16,000. The results for these cases are shown in Figures 14 and 15. It can be concluded that velocities greater than 14,500 in/s lead to perforation of the bullet with a non-zero residual velocity. The predicted V50 velocity is between 14,000 and 14,500 and underpredicts the experimental value by 16%–19%. The dynamic deflection of the center of the plate for this case is ~0.05 in.

* S-2 Glass is a registered trademark of Owens Corning Corp.

1.2 Micromechanical Material Model in LSDYNA

Tabiei and Chen developed a micromechanical material model for unidirectional composites [6]. The material model assumes linear elastic behavior up to the first constituent failure. Microfailure criteria are utilized in the model. Tabiei et. al [7] also presented a nonlinear viscoplastic strain-rate-sensitive micromechanical material model with progressive microfailure criteria. Both models are programmed as user-defined material routines in the explicit code LSDYNA. The material models are used to simulate the bullet penetration problem. The material (S-2 Glass/epoxy) properties used in the simulation are listed in Table 2. Note that in this material model, the properties of each of the constituents (fiber and resin) must be provided. Figure 16 shows the prediction of the V50 velocity of about 14,250 in/s, which underpredicts the experimental value by ~17.5%. Several computational iterations were performed to obtain this velocity, but are not illustrated in this report. Two sets of simulations are shown in Figure 16; Figure 16a is the simulations obtained with the strain-rate-sensitive micromechanical model, and Figure 16b is the simulations obtained with the strain-rate-insensitive micromechanical model. The predictions of the rate insensitive model are stiffer than those for the rate sensitive model; this can be seen from the resultant velocity, the displacement of the bullet, and the plate dynamic deflection. The increased stiffness observed for the rate insensitive micromechanical model is due to the nonlinear material behavior and strain softening for the matrix material. The dynamic deflection of the plate is predicted to be about 0.15 in, which is larger than the LSDYNA Material Model 22 prediction of 0.05 in. In both models, the effective strain is used as an ultimate criterion for element erosion. The value of the effective strain at failure is taken to be 20% for both cases. However, in the future, a strain-rate-dependent failure criterion might be employed. Note that the prediction of V50 for the rate sensitive material model is less than the prediction of the V50 with no strain rate effect. This attributed to the fact that the strain-rate-sensitive material model considers material nonlinear behavior in the composites (strain softening). However, the Material Model with no rate effect assumes that everything is elastic up to the point of failure. The principal author will be implementing a Cooper-Symond type of rate sensitive failure criterion in the referenced micromechanical model.

1.3 LAMPAT/DYNA3D Implementation

After validating the formulation for nonlinear behavior (Figures 3–5) the implemented material model is utilized to simulate the bullet penetration problem. The aim is to predict the V50 velocity using the LAMPAT/DYNA3D code. The input file for the bullet model concerning the data required for LAMPAT material subroutine is listed in Table 3. The material type is programmed as Material 46. The third row/first column requires an input of the elastic modulus. The fourth row/first column requires the Poisson's ratio. These two material constants are used in the time step calculation and the penalty contact stiffness calculation by DYNA3D. The fifth row/first column requires the interval frequency for material stiffness updates. Finally, the sixth row/first column requires the allowable for effective strain at failure for the eroding algorithm in DYNA3D. The same value (20% failure strain) is used for eroding as the other two previously presented material models. Table 4 lists the material properties used by LAMPAT/DYNA3D for the S-2 Glass/epoxy. There is difficulty defining the exponential parameters for the material nonlinear behavior in LAMPAT since they have not been determined experimentally, but for the purpose of comparison, the same exponents used in the graphite/epoxy material [2] are used for the S-2 Glass/epoxy. Two extreme cases are considered for the simulation of the bullet penetration problem. In the first simulation (case 1) the composite plate is assumed to be linear up-to-failure in LAMPAT; this behavior is achieved by suppressing the nonlinear update of the elastic and shear modulus in LAMPAT. In the second simulation (case 2), nonlinear material behavior is assumed in LAMPAT. Figures 17 and 18 show the penetration predictions of the two cases considered. In both cases, the material stiffness matrix is updated every 100 cycles. Figure 17 is for linear up-to-failure material behavior, and Figure 18 is for nonlinear up-to-failure material behavior. It is apparent that there is significant difference in the results. The nonlinear material behavior (case 2) predicts a much softer response than the prediction of the linear material behavior (case 1). For case 2, the bullet is almost intact, and for case 1 the bullet head and jacket is fully eroded. A comparison is also made for the effect of the interval of material stiffness matrix updates on the behavior of the considered problem. Figures 19 and 20 show the effective plastic strain in the bullet and the effective stress in the plate for two frequencies of the material stiffness matrix updates.

Figure 19 shows the result for every 10 cycles update, and Figure 20 shows the result for every 100 cycles update. For the 10 cycles update, the effective plastic strain in the bullet is predicted to be 0.744, and the effective stress in the plate is 245 kpsi at 60 ms. For the 100 cycles update, the effective plastic strain in the bullet is predicted to be 0.791, and the effective stress in the plate is 457 kpsi at 60 ms. The CPU time difference between the two cases is almost 1 order of magnitude. This makes the analysis with few cycles for material stiffness update uneconomical even for small-scale simulations.

4. Summary

The LAMPAT nonlinear composite material code is successfully implemented in the nonlinear dynamic explicit finite element code LLNL-DYNA3D. LAMPAT is added as a new material model in DYNA3D and is modified to fit suitably in an explicit time integration scheme. The code calls an external database file that contains properties of several composite materials. Therefore, most of the material data is read from the external file, and the limitation on the number of material constant by DYNA3D is eliminated. The implementation is verified for linear and nonlinear behavior of composites. The nonlinear behavior is verified by considering a block of material that is loaded in all three coordinate directions. Impact simulations are also carried out to investigate the stability of the LAMPAT code in the explicit finite element code DYNA3D. A bullet penetration problem is considered and several simulations are carried out using three material models. Material type 22 (laminated composite) in LSDYNA, a strain-rate-sensitive micromechanical model, and LAMPAT/DYNA3D are considered, and their predictions are presented. The following can be summarized from the presented simulation of the bullet penetration problem:

- Material Model 22 in LSDYNA underpredicted the V50 velocity by 16%–19%; however, the prediction of dynamic deflection is very small (0.03 in).
- The micromechanical material model predicted the V50 velocity comparable to Material Model 22 in LSDYNA. The dynamic deflection prediction is significantly higher than Material Model 22 (0.15 in) and seems more reasonable, although no experimental data on dynamic deflection were experimentally obtained.
- Strain-rate-sensitive and insensitive micromodels are considered for the bullet penetration problem. The strain-rate-sensitive material model considers material softening and therefore predicted a softer behavior than the rate insensitive model.
- The LAMPAT arterial nonlinear input data cannot easily be obtained; therefore, this presents difficulty for simulating penetration experiments. If material exponents similar to those in the graphite/epoxy are used in the S-2 Glass/epoxy, premature failure in the element occurs. This leads to significantly underpredicting the V50 velocity.
- The CPU time required by Material Type 22 in LSDYNA and the micromechanics material model is significantly less than that of LAMPAT. The CPU time is on the order of hours for the Material 22 and the micromechanics material model. However, it is on the order of days for the LAMPAT/DYNA3D code.
- The CPU time requirement for LAMPAT/DYNA3D increases 1 order of magnitude when the frequency of the material stiffness matrix updates is decreased by 1 order of magnitude.
- There is a significant difference in prediction between the low- and high-frequency material stiffness matrix updates. Therefore, for more accurate predictions, the material stiffness matrix must be updated more frequently. This will render analysis of even small finite element models computationally inefficient.

The LAMPAT/DYNA3D code could be improved for more accuracy in predicting impact and penetration problems. Several suggestions are presented to make the code more suitable for finite element impact simulations.

5. Possible LAMPAT Improvements

The following is a list of possible improvements that could make the LAMPAT/DYNA3D material model more accurate and robust for finite element impact simulation problems:

- Adding a prefailure unloading scheme. The current implementation assumes that unloading is along the loading path.
- Adding a postfailure unloading scheme. The current implementation also assumes that unloading is along the initial loading path.
- Adding strain rate sensitivity to the model.
- Adding a more realistic progressive failure process in the homogenization process.
- Postprocessing can be improved by identifying the various progressive failure or damage mechanisms as they evolve temporally.

6. References

1. Whirley R., and B. Engelman. "DYNA3D User's Manual." Methods Development Group, Lawrence Livermore National Laboratory, Livermore, CA, November 1993.

2. Bogetti, T., C. Hoppel, and B. Burns. "LAMPAT: A Software Tool for Analyzing and Designing Thick Laminated Composite Structures." ARL-TR-890, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, September 1995.

3. Tabiei, A., and T. Bogetti. "Implementation of the Nonlinear Composite Analysis Code "LAMPAT" Into LSDYNA." U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, to be published.

4. Hallquist, J. "LSDYNA User's Manual." Livermore Software Technology Corporation, Livermore CA, 1999.

5. Gazonas, G. A. "Computational Design and Optimization of Hybrid Laminates." Presented at the 45th Sagamore Army Materials Research Conference, St. Michaels, MD, 25–28 June 2001.

6. Tabiei, A., and Q. Chen. "Micro-Mechanics Composite Material Model for Crashworthiness Applications." *Journal of Thermoplastics Composites*, vol. 14, no. 4, pp. 264–289, July 2001.

7. Tabiei, A., Y. Weitao, and R. Goldberg. "Strain-Rate-Dependent Micro-Mechanical Composite Material Model for Impact Simulation." *International Journal of Numerical Methods in Engineering*, to be published.

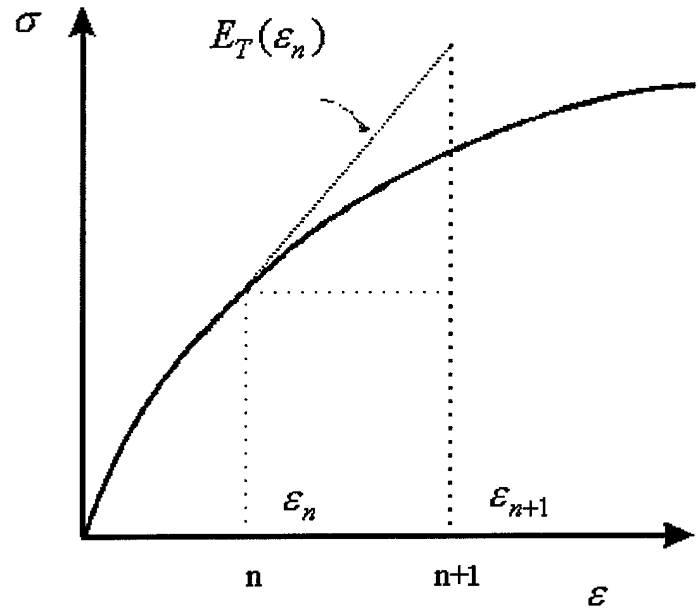


Figure 1. Load increments scheme.

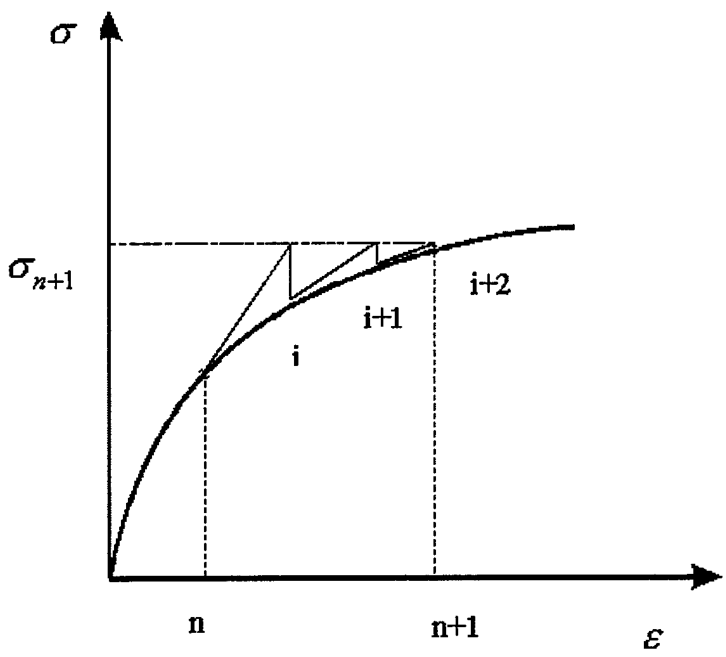


Figure 2. Equilibrium iteration scheme.

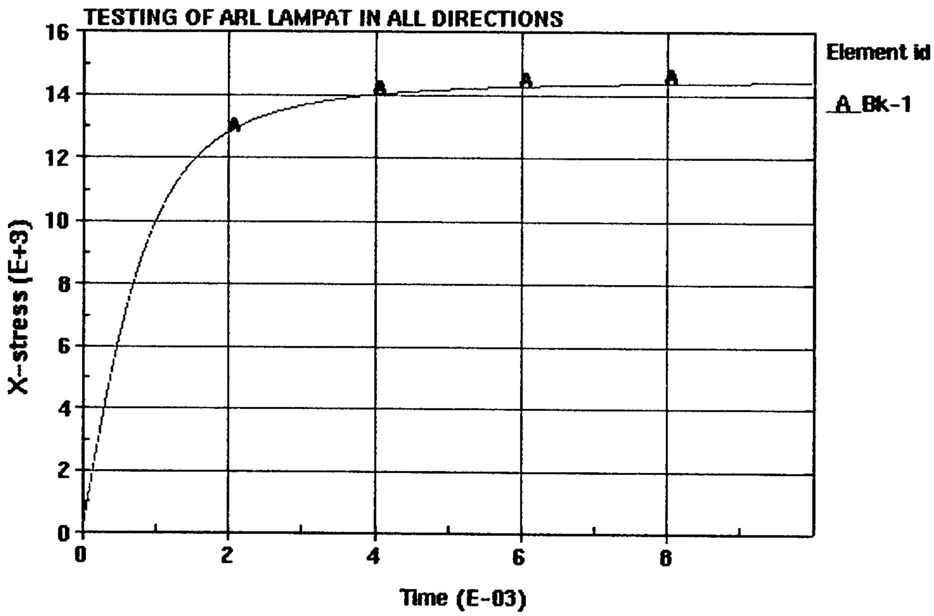


Figure 3. Material nonlinear behavior in the x-direction.

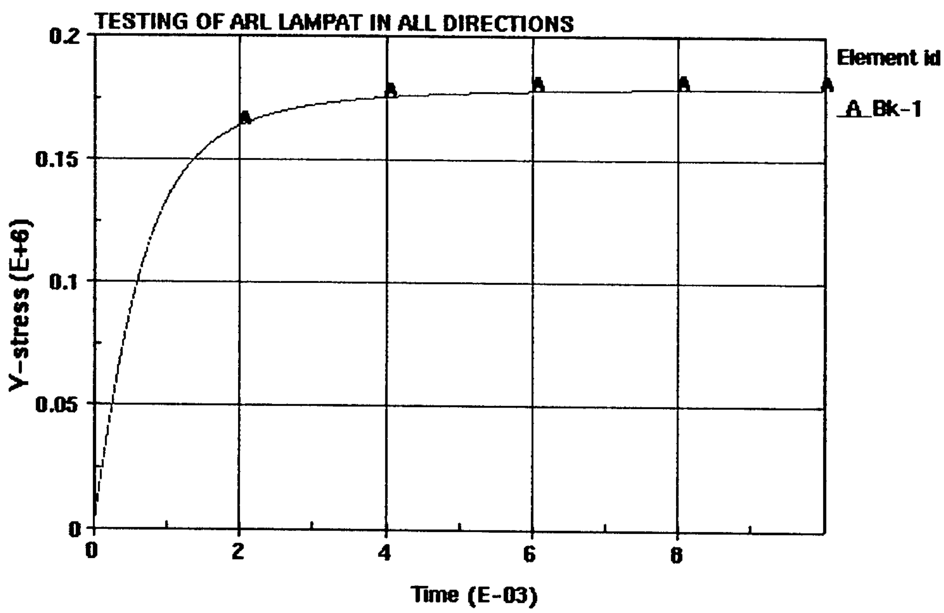


Figure 4. Material nonlinear behavior in the y-direction.

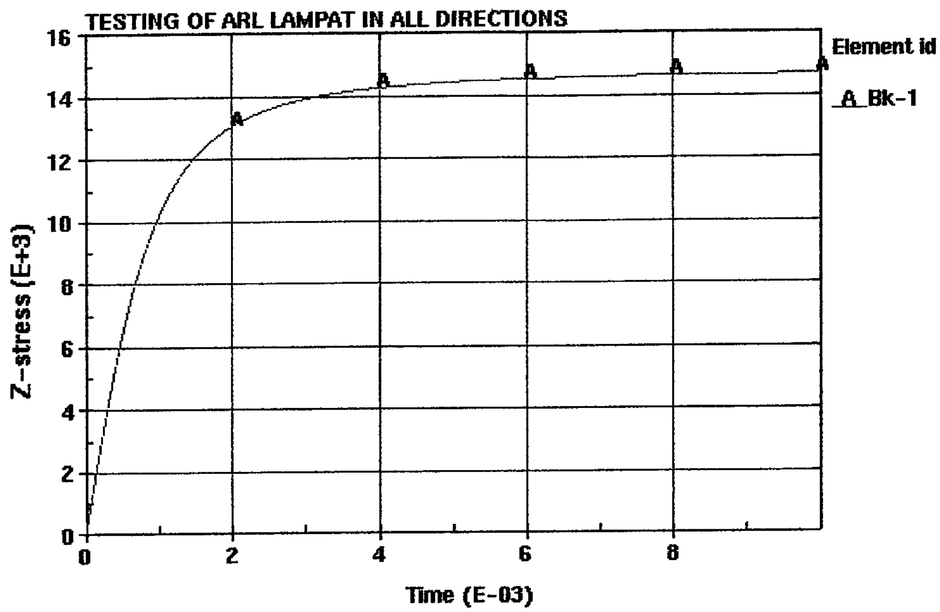


Figure 5. Material nonlinear behavior in the z-direction.

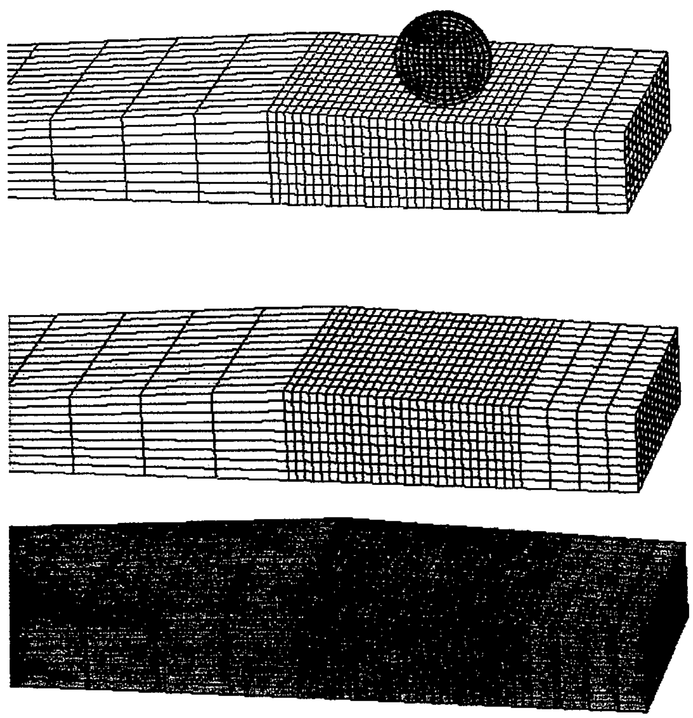


Figure 6. Penetration of a rigid ball into three composite plates (initial geometry).

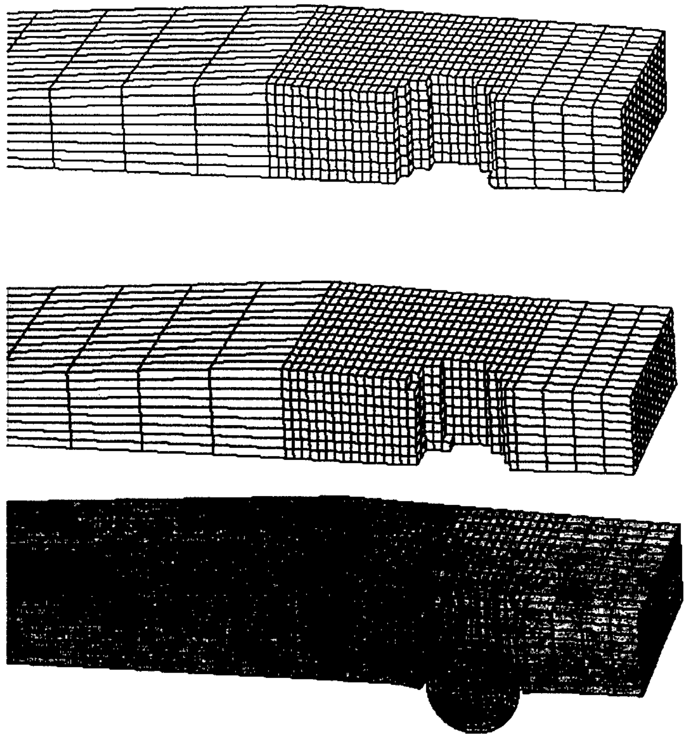


Figure 7. Penetration of a rigid ball into three composite plates (final geometry).

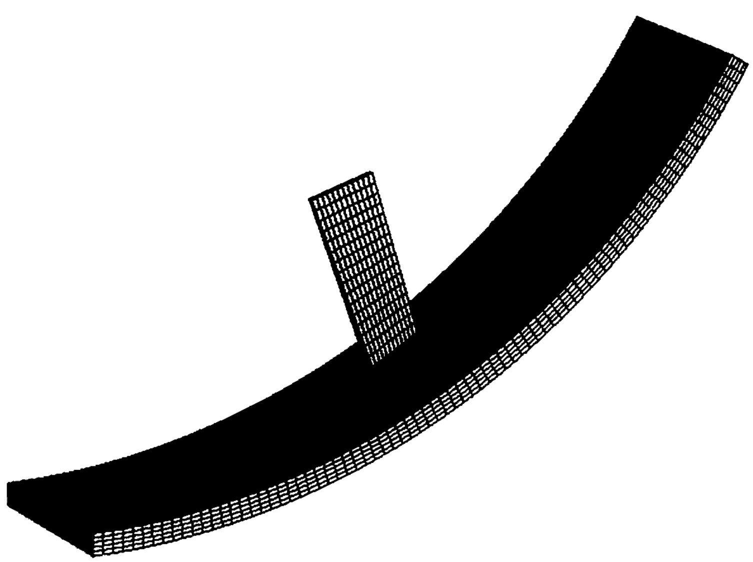


Figure 8. Penetration of a rigid plate into a composite curved plate (initial geometry).

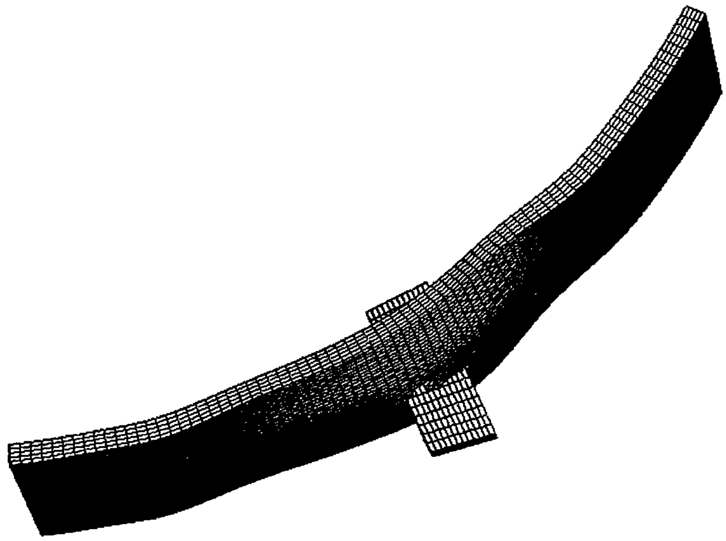


Figure 9. Penetration of a rigid plate into a composite curved plate (intermediate geometry).

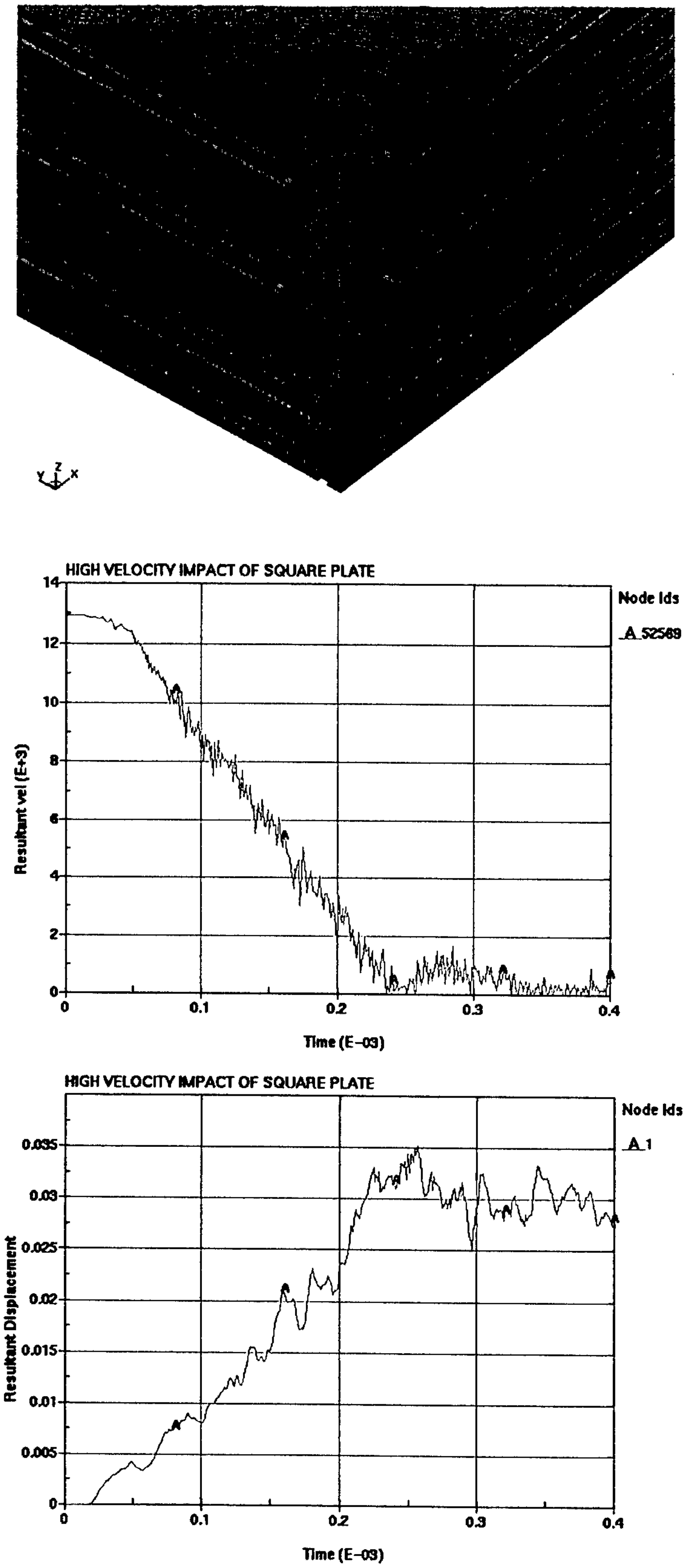


Figure 10. Impact velocity is 13,000 in/s.

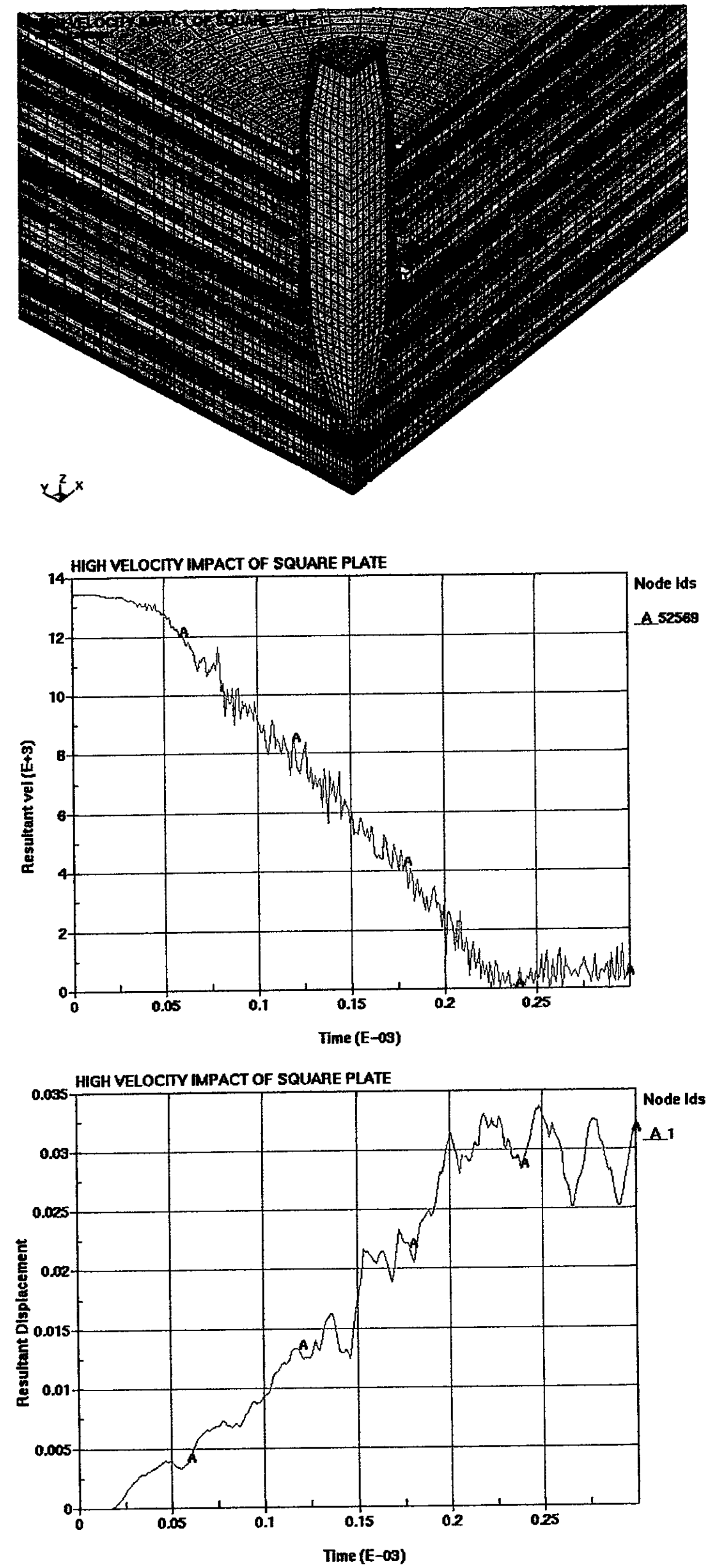
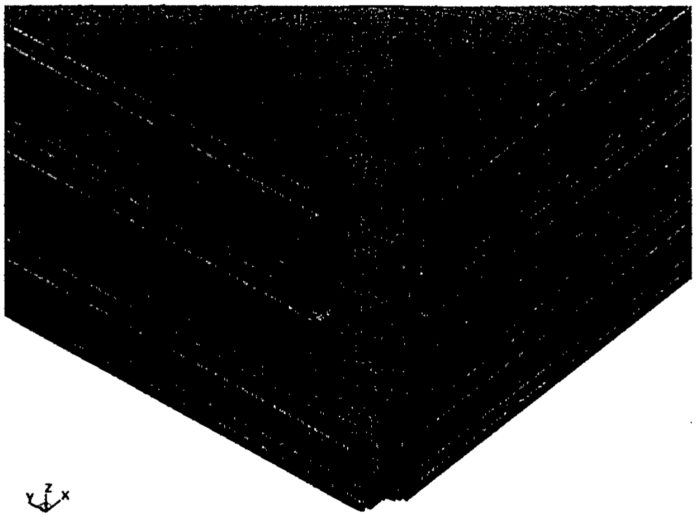
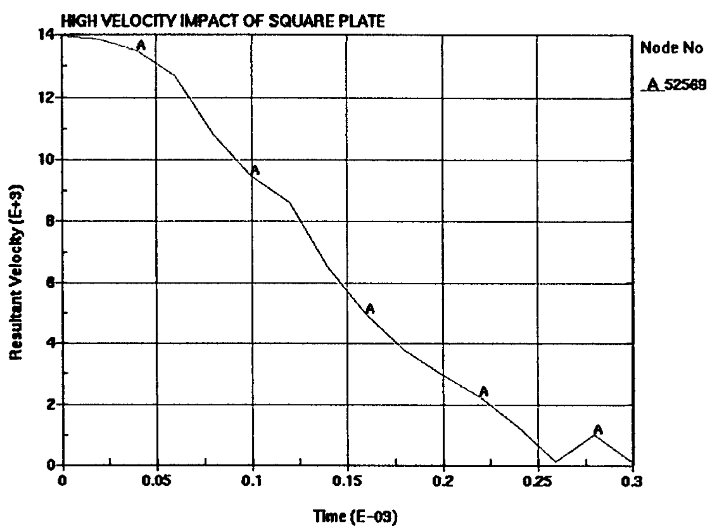


Figure 11. Impact velocity is 13,500 in/s.





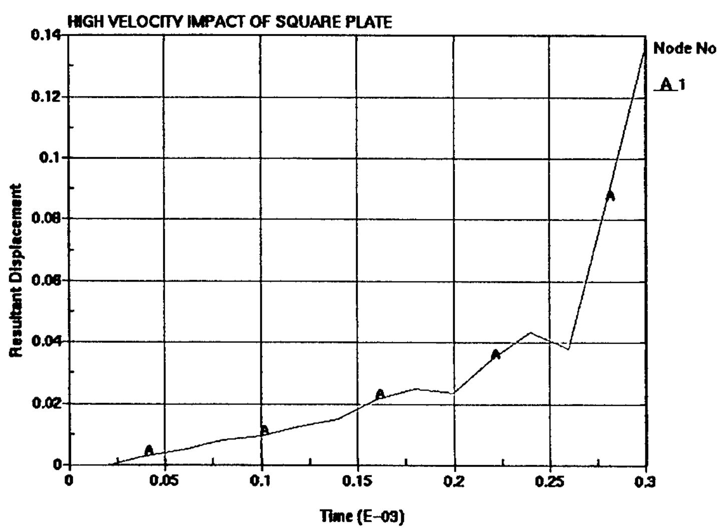
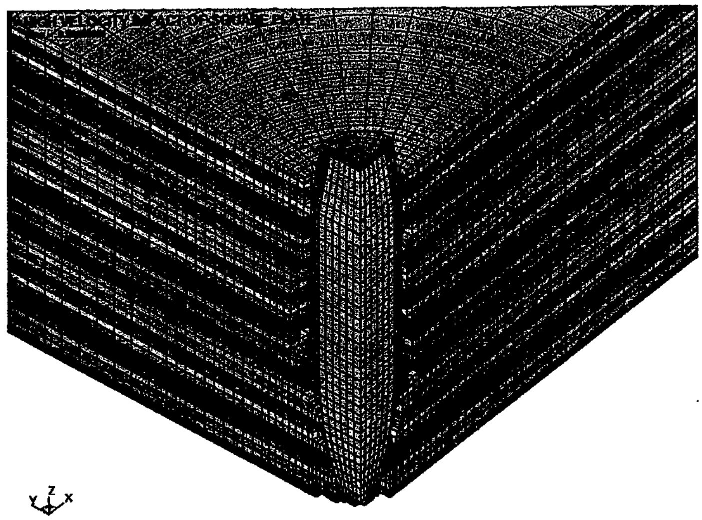
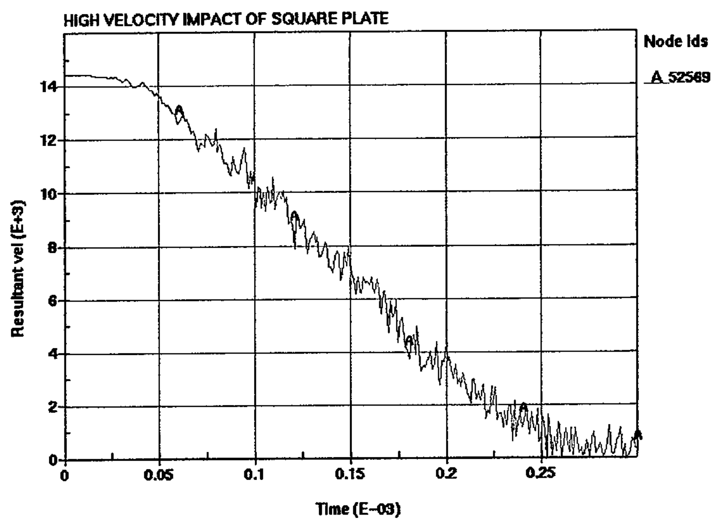


Figure 12. Impact velocity is 14,000 in/s.





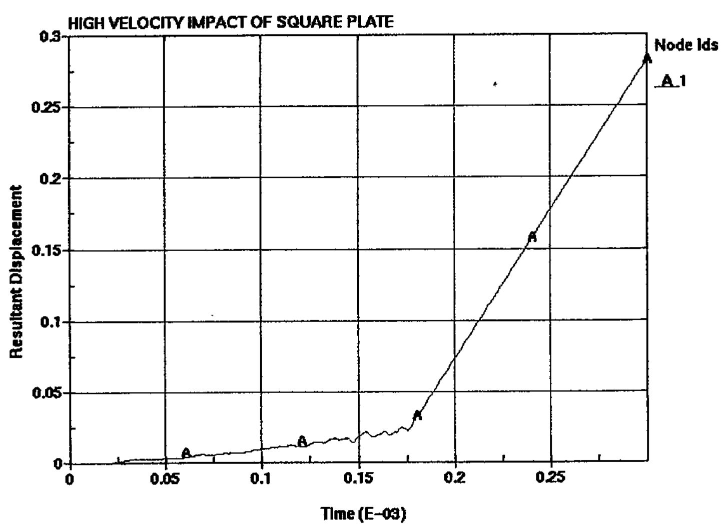


Figure 13. Impact velocity is 14,500 in/s.

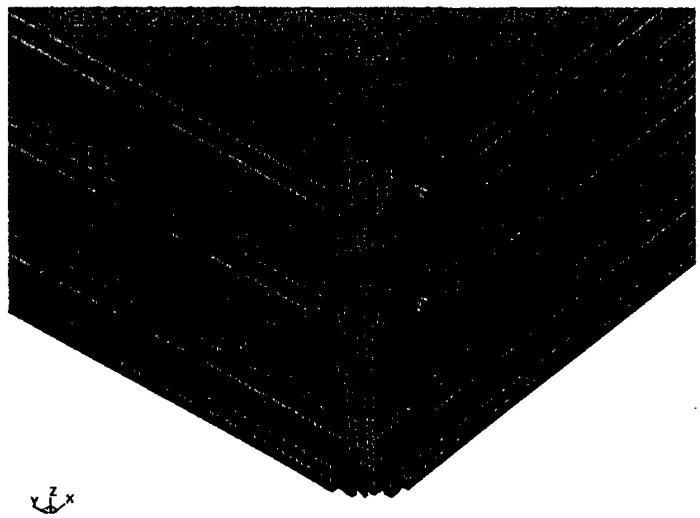


Figure 14. Impact velocity is 15,000 in/s.

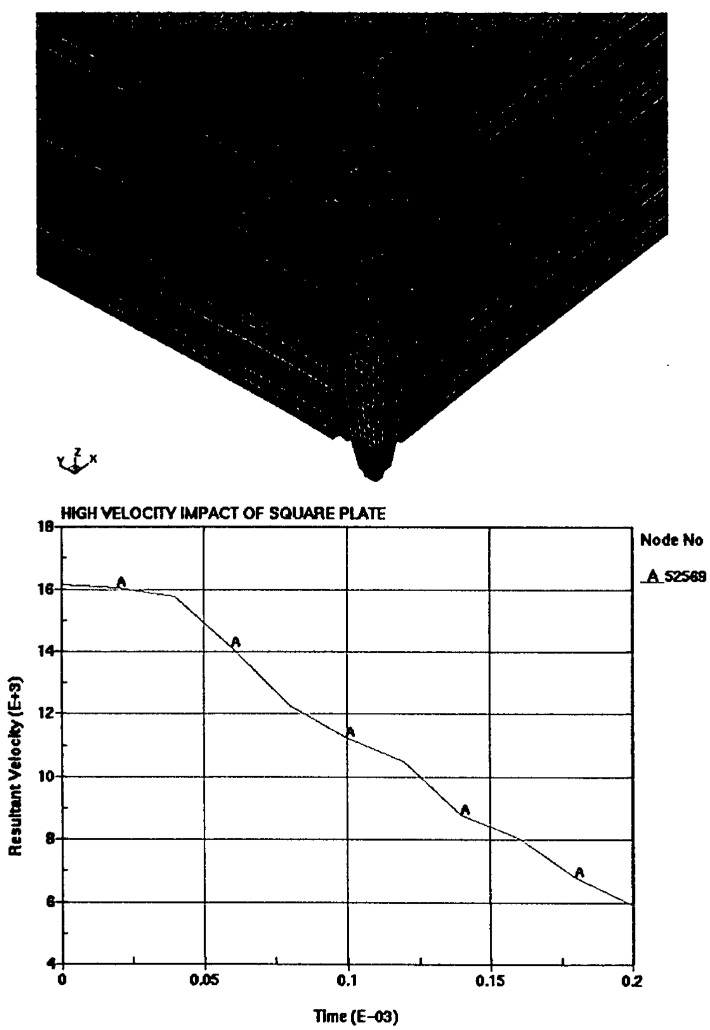
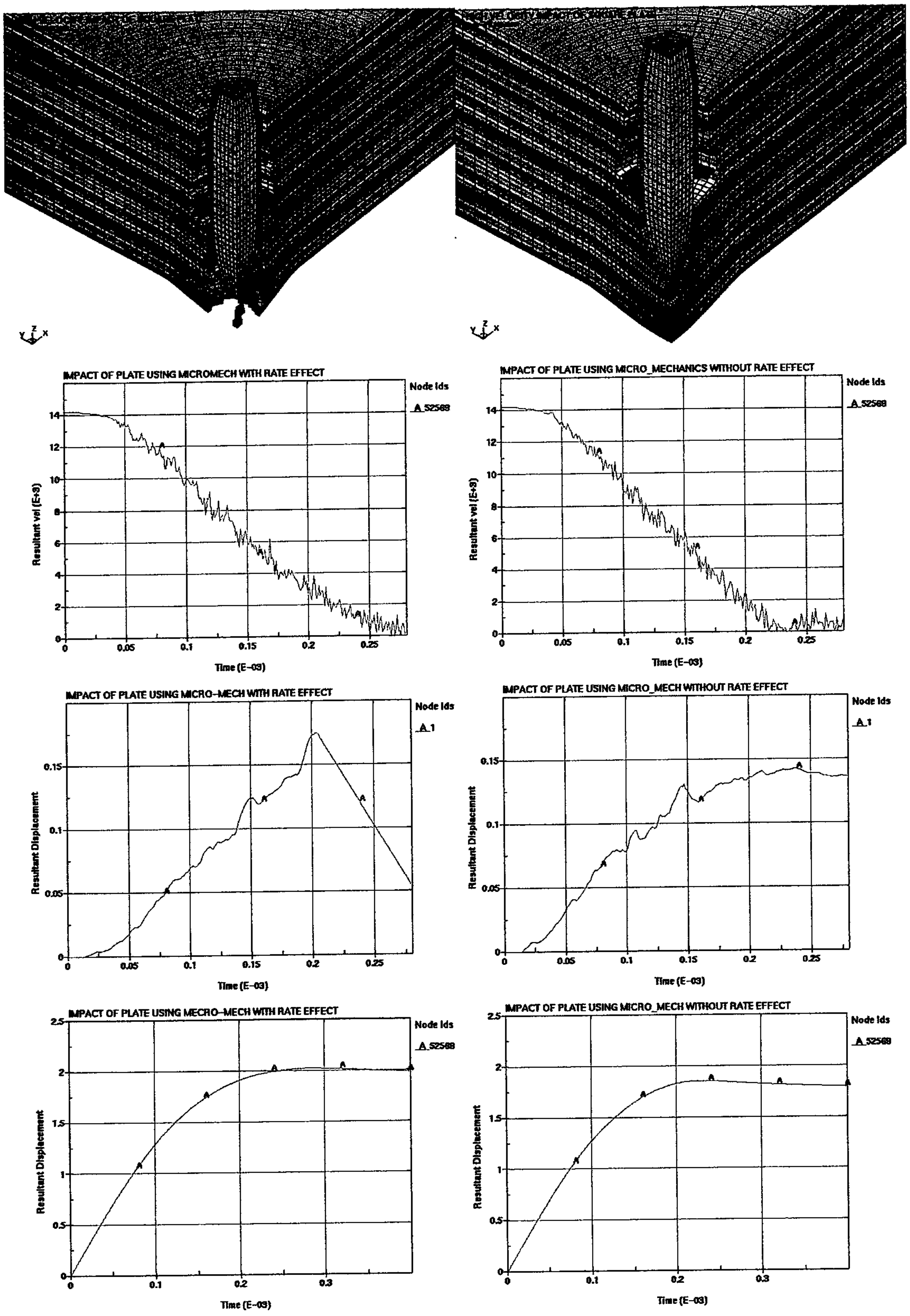


Figure 15. Impact velocity is 16,000 in/s.



(a) (b)

Figure 16. Micromechanics (a) rate sensitive and (b) rate insensitive, with impact velocity of 14,250 in/s.

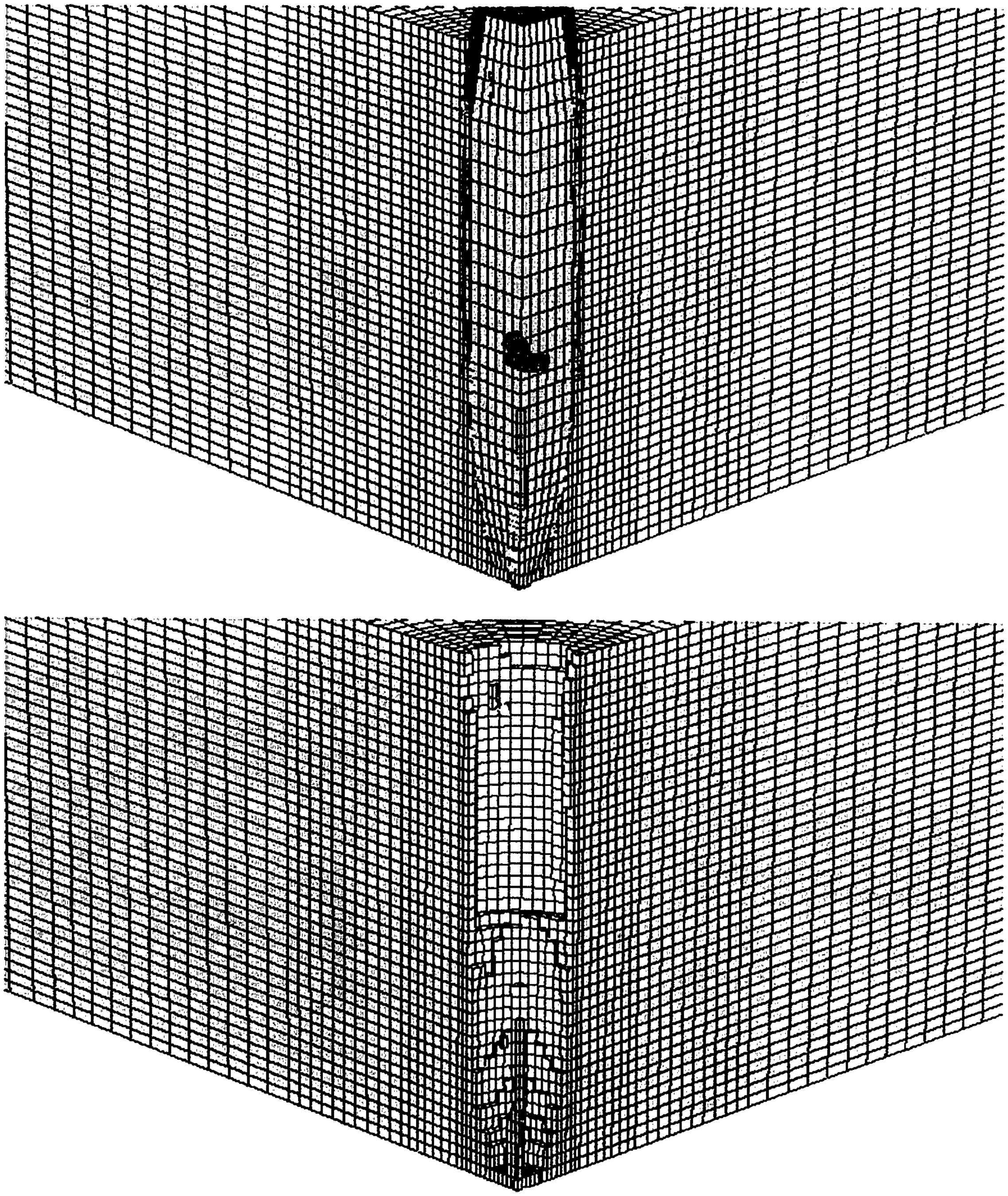


Figure 17. Linear material behavior up-to-failure.

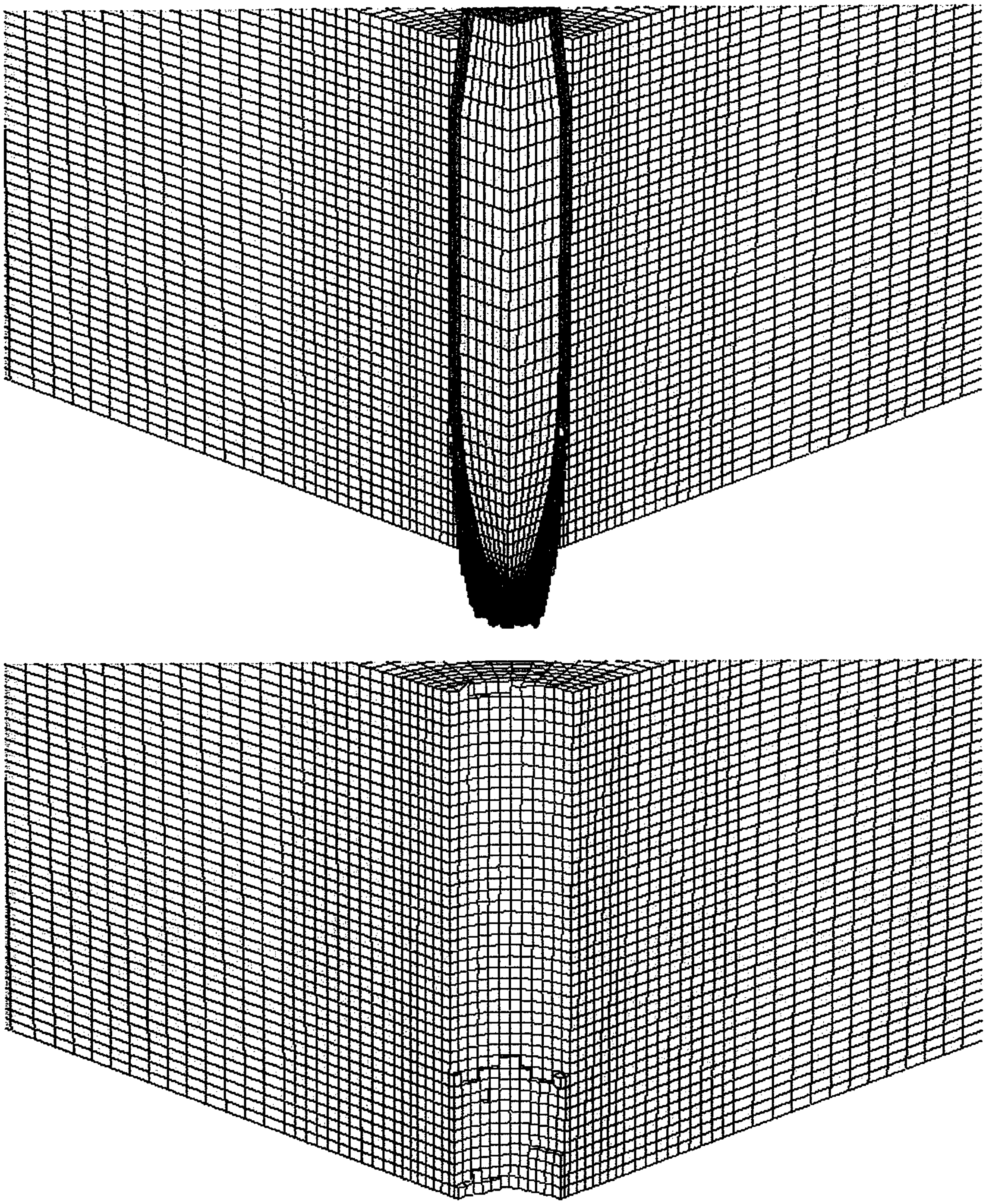


Figure 18. Nonlinear material behavior up-to-failure.

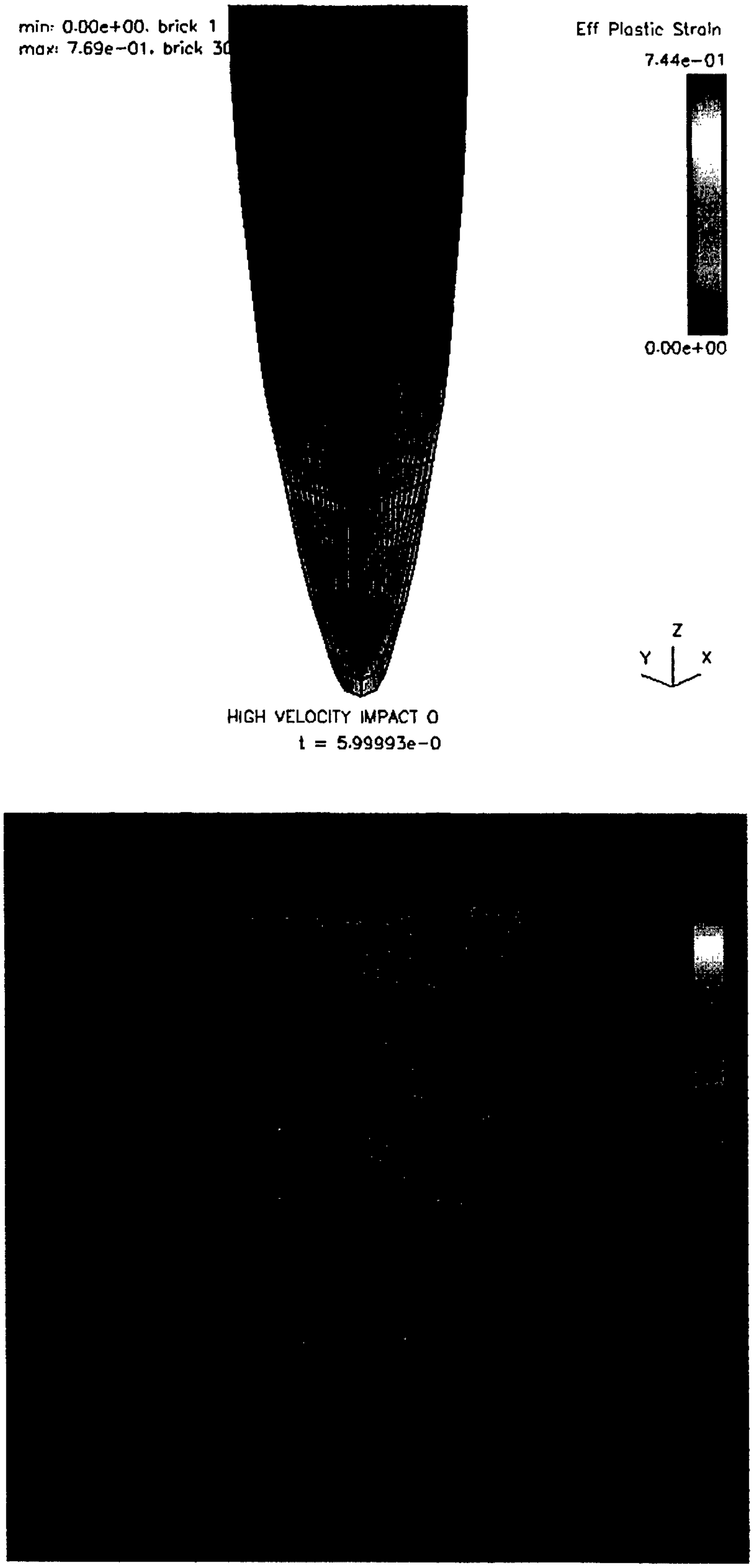


Figure 19. Stiffness update every 10 cycles.

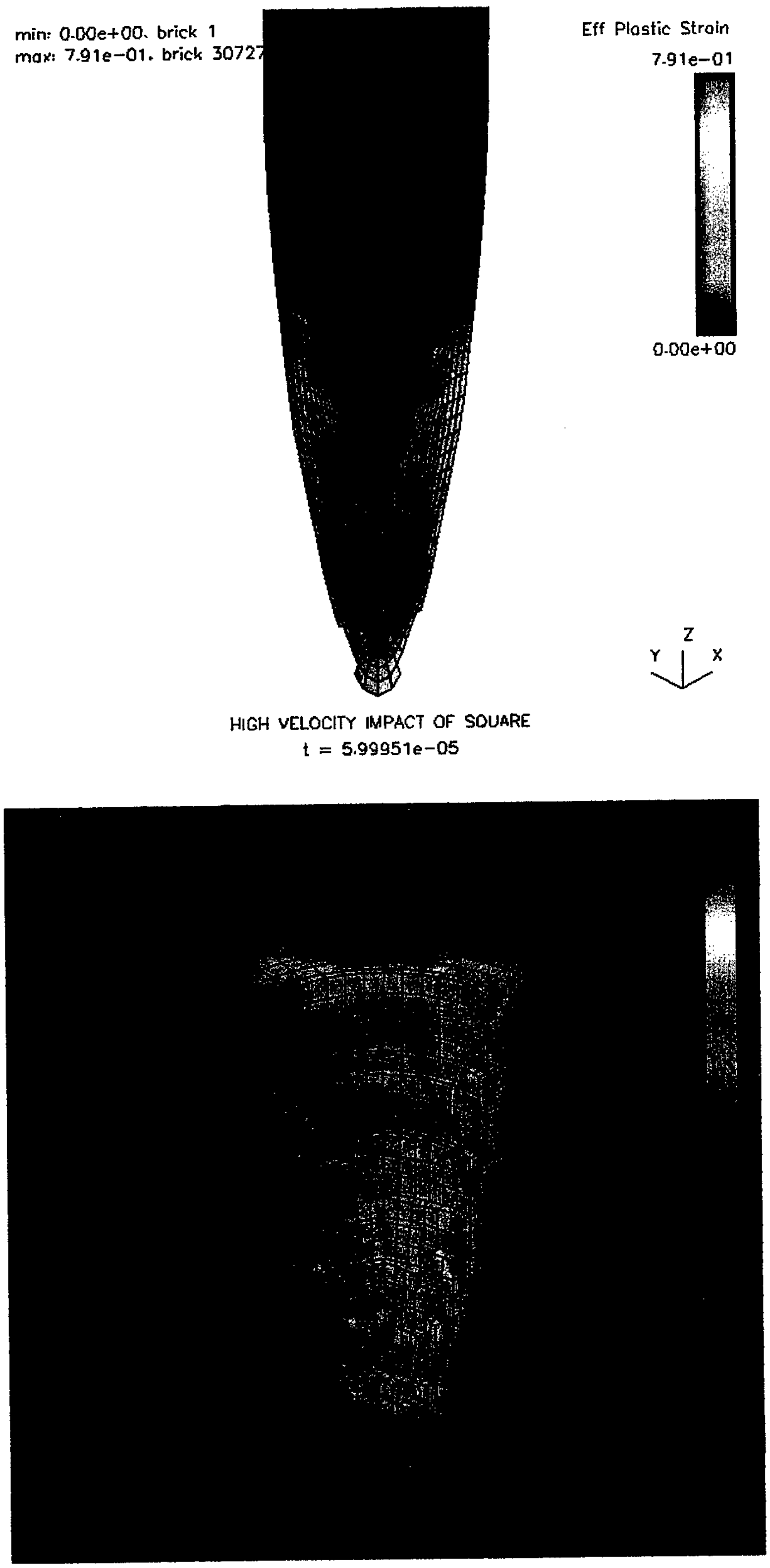


Figure 20. Stiffness update every 100 cycles.

Table 1. Material 22 in the LSDYNA code.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*MAT_COMPOSITE_DAMAGE
$ S2/Epoxy Plain Weave, 0-layer
$        i         f         f         f         f         f         f         f
$      mid        ro        eA        EB        EC      PRBA      PRCA      PRCB
         1 1.668E-04  6.00E+06  1.20E+06  1.20E+06    0.0540    0.0540      0.40
$        f         f         f         f         f         f
$      GAB       GBC       GCA     KFAIL      AOPT      MACF
  0.66E+06  0.66E+06  0.66E+06   1.0E+03       2.0       1.0
$        f         f         f         f         f         f
$       XP        YP        ZP        A1        A2        A3
       0.0       0.0       0.0       1.0       0.0       0.0
$        f         f         f         f         f         f
$       V1        V2        V3        D1        D2        D3
       0.0       0.0       0.0       0.0       1.0       0.0
$        f         f         f         f         f
$       SC        XT        YT        YC      ALPH
  12.0E+03 185.0E+03   7.1E+03  20.0E+03      0.20
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
```

Table 2. The strain-rate-sensitive micromodel as user-defined routine in LSDYNA.

```
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*MAT_USER_DEFINED_MATERIAL_MODELS
$      MID        RO        MT       LCM       NHV    IORTHO        IB        IG
         1   1.668E-4       43        21        36         1        20        21
$     IVEC     IFAIL
         0         0
$     AOPT
         2                                                   a1        a2        a3
                                                              1         0         0
$                                       d1        d2        d3
                                         0         1         0
$       Em        Gm        Vm       E1f       E2f      V12f      v23f        Gf
    0.58e6   0.206e6      0.40    12.4e6    12.4e6     0.200      0.20     5.2e6
$        f        D0         N        Z0         M    OMGMX      BETA       Xft
$     0.60     1.E04      0.70  91808.32     312.0    7614.4      0.45    600.e3
      0.60     0.E00      0.00      0.00       0.0       0.0      0.00    600.e3
$      Ytm        Sm       Ycm         B         G
     8.1e3    7.00e3   100.0e3  6888.9e3     5.2e6
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
```

Table 3. The LAMPAT material model in DYNA3D.

```
--------------------------------------------------------------------------------
   1   46 1.66800-4     0     0 0.0000000     0 0.0000000 0.0000000     0     0     0
 PART PID =          1, COMPOSITE 0
 3.00000+7 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 0.3000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 1.00000e2 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 0.2000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
--------------------------------------------------------------------------------
```

Table 4. The data file for the S-2 Glass in LAMPAT.

```
--------------------------------------------------------------------------------
         2.0,          2.0,   1.668E-04   !    MAT_ID, MAT_TYPE, MAT_DENS
(S2GLAS/EPOXY)
   6.000E+06,   1.200E+06,   1.200E+06   !        E1,        E2,        E3
   6.000E+06,   1.200E+03,   1.200E+03   !   SIG_0-1,   SIG_O-2,   SIG_O-3
        10.0,        5.00,        5.00   !       N-1,       N-2,       N-3
       0.400,       0.370,       0.370   !      NU23,      NU13,      NU12
   6.600E+05,   6.600E+05,   6.600E+05   !       G23,       G13,       G12
   7.483E+03,   7.483E+03,   7.483E+03   !  SIG_O-23,  SIG_O-13,  SIG_O-12
        2.38,        2.38,        2.38   !      N-23,      N-13,      N-12
   2.300E-06,   1.850E-05,   1.850E-05   !        A1,        A2,        A3
         0.0,         0.0,         0.0   !       VM1,       VM2,       VM3
         0.0,         0.0,         0.0   !       VM4,       VM5,       VM6
         0.0,         0.0,         0.0   !       VM7,       VM8,       VM9
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       CA1,       CA2,       CA3
         0.0,         0.0,         0.0   !       CA4,       CA5,       CA6
         0.0,         0.0,         0.0   !       CA7,       CA8,       CA9
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       CB1,       CB2,       CB3
         0.0,         0.0,         0.0   !       CB4,       CB5,       CB6
         0.0,         0.0,         0.0   !       CB7,       CB8,       CB9
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       ---,       ---,       ---
  243.00E+03,    7.00E+03,    8.50E+03   !       X1T,       X2T,       X3T
  177.00E+03,   30.60E+03,   35.00E+03   !       X1C,       X2C,       X3C
   15.70E+03,   17.00E+03,   15.70E+03   !       X23,       X13,       X12
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       ---,       ---,       ---
      0.0340,      0.0033,      0.0040   !       Y1T,       Y2T,       Y3T
      0.0248,      0.0144,      0.0164   !       Y1C,       Y2C,       Y3C
      0.0221,      0.0173,      0.0160   !       Y23,       Y13,       Y12
         0.0,         0.0,         0.0   !       ---,       ---,       ---
         0.0,         0.0,         0.0   !       ---,       ---,       ---
```

INTENTIONALLY LEFT BLANK.

Appendix A. Brick Element Formulation in DYNA3D

This appendix appears in its original form, without editorial change.

INTENTIONALLY LEFT BLANK.

Brick Element Formulation in DYNA3D

The following is a brief description of the explicit time integration algorithm in DYNA3D for *Solid Element.* This algorithm is reported in here to indicate the hierarchy and position of material model in the solid element formulation.

1. Set initial conditions at time t=0:

   $$n = 0; \Delta t = 0(\text{timestep}); X^0 = X(t=0); u^0 = u(t=0); v^{-1/2} = v(t=0).$$

   where n is the number of iteration (number of cycle).
2. Go to step 3.5.
3. Explicit time integration

   3.1 Calculate nodal accelerations

   $$a^{n-1} = \frac{1}{m_{ii}}\left[F_{ext}^{n-1} + F_{hg}^{n-1} - F_{\text{int}}^{n-1} - Cv^{n-3/2}\right]$$

   where m_{ii} is the nodal mass of node i. And C is the damping matrix.

   3.2 Impose boundary conditions

   3.3 Impose nodal constraints

   3.4 Update velocities and geometry

   $$v^{n-1/2} = v^{n-3/2} + a^{n-1}\Delta t$$

   $$u^{n} = u^{n-1} + v^{n-1/2}\Delta t$$

   $$X^{n} = X^{0} + u^{n}$$

   3.5 Compute external nodal forces F_{ext}^{n}

   - evaluate the load curves
   - compute surface tractions
   - compute concentrated loads
   - compute body force loads

   3.6 Process ***solid*** elements

   3.6.1 Compute velocity strains and spin rates

   a). Compute Hourglass Type 1, 2 or 4

   - Compute the Jacobian matrix $J(0,0,0)$, the element volume V_e, and

   $$\frac{\partial\phi_k}{\partial x_j}(k = 1,2,3,4):$$

   $$J(0,0,0) = \begin{bmatrix} \sum_{k=1}^{8} x^k\left(\frac{\partial\phi_k}{\partial\xi}\right)_{(0,0,0)} & \sum_{k=1}^{8} y^k\left(\frac{\partial\phi_k}{\partial\xi}\right)_{(0,0,0)} & \sum_{k=1}^{8} z^k\left(\frac{\partial\phi_k}{\partial\xi}\right)_{(0,0,0)} \\ \sum_{k=1}^{8} x^k\left(\frac{\partial\phi_k}{\partial\eta}\right)_{(0,0,0)} & \sum_{k=1}^{8} y^k\left(\frac{\partial\phi_k}{\partial\eta}\right)_{(0,0,0)} & \sum_{k=1}^{8} z^k\left(\frac{\partial\phi_k}{\partial\eta}\right)_{(0,0,0)} \\ \sum_{k=1}^{8} x^k\left(\frac{\partial\phi_k}{\partial\zeta}\right)_{(0,0,0)} & \sum_{k=1}^{8} y^k\left(\frac{\partial\phi_k}{\partial\zeta}\right)_{(0,0,0)} & \sum_{k=1}^{8} z^k\left(\frac{\partial\phi_k}{\partial\zeta}\right)_{(0,0,0)} \end{bmatrix}$$

   $$V_e = 8\left|J(0,0,0)\right|$$

$$\begin{bmatrix} \frac{\partial \phi_k}{\partial x} \\ \frac{\partial \phi_k}{\partial y} \\ \frac{\partial \phi_k}{\partial z} \end{bmatrix}_{(0,0,0)} = J^{-1}(0,0,0) \begin{bmatrix} \frac{\partial \phi_k}{\partial \xi} \\ \frac{\partial \phi_k}{\partial \eta} \\ \frac{\partial \phi_k}{\partial \zeta} \end{bmatrix}_{(0,0,0)} \qquad \text{(k=1, 2, 3, 4)}$$

- Compute the velocity gradient matrix **L**, velocity strains $\dot{\varepsilon}$ and spin rates ω:

$$L_{ij}^{n-1/2} = \frac{\partial v_i^{n-1/2}}{\partial x_j} = \sum_{k=1}^{8} (v_i^k)^{n-1/2} \left(\frac{\partial \phi_k}{\partial x_j} \right)_{(0,0,0)}$$

$$= \left[(v_i^1)^{n-1/2} - (v_i^7)^{n-1/2} \right] \left(\frac{\partial \phi_1}{\partial x_j} \right)_{(0,0,0)} + \left[(v_i^2)^{n-1/2} - (v_i^8)^{n-1/2} \right] \left(\frac{\partial \phi_2}{\partial x_j} \right)_{(0,0,0)}$$

$$+ \left[(v_i^3)^{n-1/2} - (v_i^5)^{n-1/2} \right] \left(\frac{\partial \phi_3}{\partial x_j} \right)_{(0,0,0)} + \left[(v_i^4)^{n-1/2} - (v_i^6)^{n-1/2} \right] \left(\frac{\partial \phi_4}{\partial x_j} \right)_{(0,0,0)}$$

$$(i, j = 1,2,3)$$

$$\dot{\varepsilon}^{n-1/2} = \frac{1}{2} [L^{n-1/2} + (L^{n-1/2})^T]$$

$$\omega^{n-1/2} = \frac{1}{2} [L^{n-1/2} - (L^{n-1/2})^T]$$

$$\omega^{n-1/2} \Delta t = \frac{\Delta t}{2} [L^{n-1/2} - (L^{n-1/2})^T]$$

- Compute $\dot{\varepsilon}_{ii}$ for the timestep calculation.

b). Compute Hourglass Type 3 or 5

3.6.2 Compute the characteristic length l_e for the timestep calculation

$$l_e = \frac{V_e}{A_{e_{max}}}$$

where V_e is the element volume, $A_{e_{max}}$ is the area of the largest side.

3.6.3 Constitutive Modeling

a). Get the global stresses σ^{n-1}_{global} from storage

b). Compute $[\sigma^{n-1} - \sigma^{n-1}(\omega^{n-1/2}\Delta t) + (\omega^{n-1/2}\Delta t)\sigma^{n-1}]_{global}$

c). Stress calculation

- Find local coordinate system and transformation matrix.

Find the unit vectors of local coordinate system:

$$\mathbf{e}_3 = \frac{\mathbf{r}_{31} \times \mathbf{r}_{42}}{\|\mathbf{r}_{31} \times \mathbf{r}_{42}\|}; \quad \mathbf{e}_1 = \frac{\mathbf{r}_{21} - (\mathbf{r}_{21} \cdot \mathbf{e}_3)\mathbf{e}_3}{\|\mathbf{r}_{21} - (\mathbf{r}_{21} \cdot \mathbf{e}_3)\mathbf{e}_3\|}; \quad \mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1.$$

where $\mathbf{r}_{ij}$ denotes the vector from Node *j* to Node *i*.

Assume $\mathbf{e}_i = l_i\hat{\mathbf{i}} + m_i\hat{\mathbf{j}} + n_i\hat{\mathbf{k}}$, and $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ are the unit vectors of global system, we have the transformation matrix **T**:

$$T = [\mathrm{e}_1 \quad \mathrm{e}_2 \quad \mathrm{e}_3]^T = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix}$$

- Transform the global stresses $[\sigma^{n-1} - \sigma^{n-1}(\omega^{n-1/2}\Delta t) + (\omega^{n-1/2}\Delta t)\sigma^{n-1}]_{\text{global}}$ to local system:

$$[\sigma^{n-1} - \sigma^{n-1}(\omega^{n-1/2}\Delta t) + (\omega^{n-1/2}\Delta t)\sigma^{n-1}]_{local}$$
$$= T\left([\sigma^{n-1} - \sigma^{n-1}(\omega^{n-1/2}\Delta t) + (\omega^{n-1/2}\Delta t)\sigma^{n-1}]_{global} T^T\right)$$

- Transform the global velocity strains $\dot{\varepsilon}^{n-1/2}$ to local system:

$$\left(\dot{\varepsilon}^{n-1/2}\right)_{local} = T\left[\left(\dot{\varepsilon}^{n-1/2}\right)_{global} T^T\right]$$

- Compute the value of $\max\{C_{11}, C_{22}, C_{33}\}$ from the material matrix for the timestep calculation.
- Update stresses in local system

$$\left(\sigma_{ij}^{n}\right)_{local} = [\sigma_{ij}^{n-1} - \sigma_{ik}^{n-1}(\omega_{kj}^{n-1/2}\Delta t) + (\omega_{ik}^{n-1/2}\Delta t)\sigma_{kj}^{n-1}]_{local} + C_{ijkl}\left(\dot{\varepsilon}_{kl}^{n-1/2}\right)_{local} \Delta t$$

- Transform the stresses σ^n from local system to global system

$$\sigma^{n}_{global} = T^T(\sigma^{n}_{local} T)$$

d). Timestep calculation

- Compute the adiabatic sound speed

$$c = \sqrt{\frac{\max\{C_{11}, C_{22}, C_{33}\}}{\rho}}$$, where ρ is the material density.

- Compute q which is a function of the bulk viscosity coefficients C_0 and C_1:

$$q = \begin{cases} C_1 c + C_0 l_e |\dot{\varepsilon}_{ii}| & \text{for } \dot{\varepsilon}_{ii} < 0 \\ 0 & \text{for } \dot{\varepsilon}_{ii} \geq 0 \end{cases}$$

where l_e is the characteristic length of the solid element.

- Compute the timestep for each element

$$\Delta t_e = \frac{l_e}{q + (q^2 + c^2)^{1/2}}$$

and take the minimum value

$\Delta t = \min\{\Delta t_1, \Delta t_2, ..., \Delta t_N\}$, where N is the number of elements.

e). Energy calculation

f). Put the global stresses σ^{n}_{global} in storage

3.6.4 Calculation of Hourglass forces and internal forces ($F^n_{hg} - F^n_{int}$)

- Calculate Hourglass forces for elements f^n_{hg}
- Calculate nodal internal forces of elements f^n_{int}

$$f_{int}^{n} = \int_{V} B^{T} \sigma dV = B^{T}(0,0,0)\sigma^{n} V_{e}$$

- Assemble $(f_{hg}^{n} - f_{int}^{n})$ into global nodal force vector $(F_{ext}^{n} + F_{hg}^{n} - F_{int}^{n})$

3.7 Prepare to start the next step

$n = n + 1$

3.8 Go to step 3.1

The above is a detailed flow chart of all steps necessary to understand and program the LAMPAT within the solid element. As can be seen from the above section (or bullet) 3.6.3 is where the modification for the material model enters the formulation.

Appendix B. LAMPAT FORTRAN Source Code

This appendix appears in its original form, without editorial change.

INTENTIONALLY LEFT BLANK.

LAMPAT/DYNA3D FORTRAN CODE

```
      subroutine f3dm46 (cm,bqs,nhex)
      parameter(lnv=32)
      implicit double precision (a-h,o-z)
dp
c
c     LAMPAT IN DYNA3D, By: A. TABIEI (ala.tabiei@uc.edu), Feb. 2001
c
      common/bk02/dt1,dt2,iburn,isdo,iorder
      common/bk06/time(2,8),head(12),idmmy,iadd,ifil,maxsiz,ncycle
      common/bk20/idummy(10),ndum
      common/aux2/d1(lnv),d2(lnv),d3(lnv),d4(lnv),d5(lnv),d6(lnv),
     1 wzzdt(lnv),wyydt(lnv),wxxdt(lnv),einc(lnv)
      common/aux14/
     + sig1(lnv),sig2(lnv),sig3(lnv),sig4(lnv),
     + sig5(lnv),sig6(lnv),epx(lnv),davg(lnv),
     + hisv1(lnv),hisv2(lnv),hisv3(lnv),hisv4(lnv),hisv5(lnv),
     + hisv6(lnv),
     + hisv7(lnv),hisv8(lnv),hisv9(lnv),hisv10(lnv),hisv11(lnv),
     + hisv12(lnv),hisv13(lnv),hisv14(lnv),hisv15(lnv),hisv16(lnv),
     + hisv17(lnv),hisv18(lnv),hisv19(lnv),hisv20(lnv),hisv21(lnv),
     + hisv22(lnv),hisv23(lnv),hisv24(lnv),hisv25(lnv),hisv26(lnv),
     + hisv27(lnv),hisv28(lnv),hisv29(lnv),hisv30(lnv),hisv31(lnv),
     + hisv32(lnv),hisv33(lnv),hisv34(lnv),hisv35(lnv),hisv36(lnv),
     + hisv37(lnv),hisv38(lnv),hisv39(lnv),hisv40(lnv),hisv41(lnv),
     + hisv42(lnv),hisv43(lnv),hisv44(lnv),hisv45(lnv),hisv46(lnv),
     + hisv47(lnv),hisv48(lnv),hisv49(lnv),hisv50(lnv),hisv51(lnv),
     + hisv52(lnv),hisv53(lnv),hisv54(lnv),hisv55(lnv),hisv56(lnv),
     + hisv57(lnv),hisv58(lnv),hisv59(lnv),hisv60(lnv),hisv61(lnv),
     + hisv62(lnv),hisv63(lnv),hisv64(lnv),hisv65(lnv),hisv66(lnv),
     + hisv67(lnv),hisv68(lnv),hisv69(lnv),hisv70(lnv),hisv71(lnv),
     + hisv72(lnv),hisv73(lnv),hisv74(lnv),hisv75(lnv),hisv76(lnv),
     + hisv77(lnv),hisv78(lnv),hisv79(lnv),hisv80(lnv),hisv81(lnv),
     + hisv82(lnv),hisv83(lnv),hisv84(lnv),hisv85(lnv),hisv86(lnv),
     + hisv87(lnv),hisv88(lnv),hisv89(lnv),hisv90(lnv),hisv91(lnv),
     + hisv92(lnv),hisv93(lnv),hisv94(lnv),hisv95(lnv),hisv96(lnv),
     + hisv97(lnv),hisv98(lnv),hisv99(lnv)
      common/aux18/dd(lnv),dfe(lnv)
      common/aux33/ix1(lnv),ix2(lnv),ix3(lnv),ix4(lnv),ix5(lnv),
     1             ix6(lnv),ix7(lnv),ix8(lnv),mxt(lnv),nmel
      common/aux35/rhoa(lnv),cb(lnv),p(lnv)
      common/aux36/lft,llt
      common/failu/sieu(lnv),failu(lnv)
      common/sandb/isandb,nacbug,issflg
      common /amain/ipblank
      pointer(ipblank,a(1))
      dimension  cm(*),bqs(*),nhex(*),effs(lnv)
c
        REAL N_E,N_S
        INTEGER I,J
        INTEGER NREG,NMAT
        DIMENSION DREG(103,4,5),DMAT(34,3,20),
     +  EE(3,100),SIG_OE(3,100),N_E(3,100),
     +  GG(3,100),SIG_OS(3,100),N_S(3,100),UEE(3,100),
     +  ALPHA(6,100),TH(100),ANGD(100),DDT(100),
```

```
     +   HV(100+1),ZV(100),FR(5,3,100),
     +   CTSIG(6,6),CTEPS(6,6),CTSIGI(6,6),CTEPSI(6,6),
     +   EPSL(6),EPSG(6,100),EPSP(6,100),
     +   ET(3,100),GT(3,100),
     +   CIJP(6,6,100),CIJB(6,6,100),ALPHAB(6,100),
     +   CCM(6,6),CMNEW(6,6),AB(6,6),
     +   SIGL(6),RNU(3,100),DEPSL(6),DSIGL(6),
     +   TEMP1(6),TEMP2(6),TEMP3(6)
         LOGICAL KFIRST
         DATA KFIRST/.FALSE./
         SAVE KFIRST
       OPEN(UNIT=99,FILE='/home/ucn582/workarl/data.in'
     +,STATUS='OLD')
c         OPEN(UNIT=1,FILE='out',STATUS='old')
c
c         write(*,*)'cycles', ncycle
         DO 45 I=1,6
         EPSL(I)=0.0
         TEMP2(I)=0.0
         TEMP3(I)=0.0
         DO 45 J=1,6
            CMNEW(I,J) = 0.0
             CTSIG(i,j)=0.0
             CTEPS(i,j)=0.0
             CTSIGI(i,j)=0.0
             CTEPSI(i,j)=0.0
45       CONTINUE
         DO 46 I=1,6
         DO 46 J=1,INPLY
         EPSG(I,J)=0.0
         EPSP(I,J)=0.0
46      CONTINUE
c
c
        IF (KFIRST) GOTO 900
c
      do 999 i=1,6
      do 999 j=1,6
      CCM(i,j)=0.0
999   continue
        READ(99,*)          NREG
        READ(99,*)          NMAT
      write(1,*)'NREG=',NREG,'   NMAT=',NMAT
      DO 12 IIA=1,NREG
      write(1,*)'---------------------------------------'
      DO 22 IC=1,2
        READ(99,*)  (DREG(IC,IB,IIA),IB=1,4)
        write(1,*)  (DREG(IC,IB,IIA),IB=1,4)
22    CONTINUE
        READ(99,*)  (DREG(3,IB,IIA),IB=1,4)
        write(1,*)  (DREG(IC,IB,IIA),IB=1,4)
        IND2=3.0+DREG(3,1,IIA)
      DO 32 IC=4,IND2
        READ(99,*)  (DREG(IC,IB,IIA),IB=1,4)
        write(1,*)  (DREG(IC,IB,IIA),IB=1,4)
```

```
32      CONTINUE
12      CONTINUE
        NLINES = 34
        DO 10 IIA=1,NMAT
        write(1,*)'MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM'
        DO 10 IC=1,NLINES
          READ(99,*)  (DMAT(IC,IB,IIA),IB=1,3)
          write(1,*)  (DMAT(IC,IB,IIA),IB=1,3)
10      CONTINUE
        write(1,*)'================================================='
c
        CLOSE(99)
        KFIRST = .TRUE.
900     CONTINUE
c
        mx=48*(mxt(lft)-1)
        ym=cm(mx+1)
        pr=cm(mx+6)
        pass=cm(mx+16)
        fs=cm(mx+21)
c        write(*,*)'fs',fs
        ss=cm(mx+2)
        g=ym/(1.+pr)
        gdt=dt1*g
        gd2=.5*gdt
        blk=-dt1*ym/((1.-2.*pr))
c
        do 910 i=lft,llt
        cb(i)=ss
        davg(i)=third*dd(i)
        p(i)=blk*davg(i)
        einc(i)=(d1(i)*sig1(i)+d2(i)*sig2(i)+d3(i)*sig3(i)+d4(i)*sig4(i)
     1          +d5(i)*sig5(i)+d6(i)*sig6(i)+dd(i)*bqs(i))*dt1
  910 continue
c
c        write(*,*)'lft= ',lft,'llt= ',llt
        do 920 in=lft,llt
c
c
           IIA  =cm(mx+11)
c           write(*,*)'IIA',IIA
              FAIL_CRT   =  DREG(1,2,IIA)
                 NLOOP   =  DREG(1,3,IIA)
            BETA   =   DREG(2,1,IIA)
             PHI   =   DREG(2,2,IIA)
              SI   =   DREG(2,3,IIA)
              NPLY        =  DREG(3,1,IIA)
        DO 220 IC=1,NPLY
              IDMAT       =  DREG(IC+3,1,IIA)
c        write(*,*)'IDMAT=',IDMAT
        DO 230 J=1,3
              EE(J,IC)    =  DMAT(2,J,IDMAT)
          SIG_OE(J,IC)    =  DMAT(3,J,IDMAT)
             N_E(J,IC)    =  DMAT(4,J,IDMAT)
             RNU(J,IC)    =  DMAT(5,J,IDMAT)
              GG(J,IC)    =  DMAT(6,J,IDMAT)
          SIG_OS(J,IC)    =  DMAT(7,J,IDMAT)
```

```
          N_S(J,IC)   =  DMAT(8,J,IDMAT)
        ALPHA(J,IC)   =  DMAT(9,J,IDMAT)
          UEE(J,IC)   =  DMAT(2,J,IDMAT)
230   CONTINUE
      DO 241 J=4,6
        ALPHA(J,IC)   =  0.0
241   CONTINUE
             TH(IC)   =  DREG(IC+3,2,IIA)
           ANGD(IC)   =  DREG(IC+3,3,IIA)
            DDT(IC)   =  DREG(IC+3,4,IIA)
               IROW   =  10.0 + ( (FAIL_CRT-1.0) * 5.0 )
      DO 242 I=1,5
      DO 242 J=1,3
          FR(I,J,IC)   =  DMAT((IROW-1)+I,J,IDMAT)
242   CONTINUE
220   CONTINUE
c
c      write(*,*)'hisv99',hisv99(in),'cycle',ncycle
      if (hisv99(in).eq.0.0) goto 2000
      cnn1=hisv99(in)*pass
      cnn=ncycle
      if (cnn.lt.cnn1) goto 1000
2000  hisv99(in)=hisv99(in)+1.0
c
       TEMP1(1)=hisv1(in)
       TEMP1(2)=hisv2(in)
       TEMP1(3)=hisv3(in)
       TEMP1(4)=hisv5(in)
       TEMP1(5)=hisv6(in)
       TEMP1(6)=hisv4(in)
c
       CALL KTRANSN (BETA,PHI,SI,CTSIG,CTEPS,CTSIGI,CTEPSI)
       CALL KMATRIX (CTEPS,TEMP1,EPSL,6,6,6,6,6,6,1,3,IERR)
c
      DO 11 K=1,NPLY
        EPSG(1,K)=EPSL(1)
        EPSG(2,K)=EPSL(2)
        EPSG(3,K)=EPSL(3)
        EPSG(4,K)=EPSL(4)
        EPSG(5,K)=EPSL(5)
        EPSG(6,K)=EPSL(6)
11    CONTINUE
      BETA = 0.0
      PHI  = 0.0
      DO 123 K=1,NPLY
       SI=ANGD(K)
c       write(1,*)'////////// si////////',SI
c
       CALL KTRANSN (BETA,PHI,SI,CTSIG,CTEPS,CTSIGI,CTEPSI)
       CALL KMATRIX (CTEPS,EPSG(1,K),EPSP(1,K),
     +              6,6,6,6,6,6,1,3,IERR)
c
123     CONTINUE
513     format(6(e14.8,1x,e14.8))
c
       CALL KTANMOD (NPLY,EPSP,EE,SIG_OE,N_E,GG,SIG_OS,N_S,ET,GT)
       CALL KCIJPRNLX (NPLY,UEE,ET,RNU,GT,CIJP)
```

```
c
c      write(1,*)'****************CIJP************************'
c      do 197 i=1,6
c      write(1,511) (CIJP(i,j,1),j=1,6)
c197    continue
c
       CALL KBARS (NPLY,ANGD,CIJP,ALPHA,CIJB,ALPHAB)
       CALL KTHICK (NPLY,TH, HV,ZV,TOT)
       CALL KCHOU (NPLY,CIJB,TH,TOT, CCM)
c
          BETA  =   DREG(2,1,IIA)
           PHI  =   DREG(2,2,IIA)
            SI  =   DREG(2,3,IIA)
c      write(1,*) 'beta,phi,si',BETA,PHI,SI
c      write(1,*)'****************CIJB************************'
c      do 194 i=1,6
c      write(1,511) (CIJB(i,j,1),j=1,6)
c194    continue
c
        CALL KTRANSN (BETA,PHI,SI,CTSIG,CTEPS,CTSIGI,CTEPSI)
        CALL KMATRIX (CCM,CTEPS,AB,6,6,6,6,6,6,6,3,IERR)
c
c      write(1,*)'++++++++++++++CTEPS+++++++++++++++'
c      do 195 i=1,6
c      write(1,511) (CTEPS(i,j),j=1,6)
c195    continue
c
        CALL KMATRIX (CTSIGI,AB,CMNEW,6,6,6,6,6,6,6,3,IERR)
c
      hisv7(in)=CMNEW(1,1)
      hisv8(in)=CMNEW(1,2)
      hisv9(in)=CMNEW(1,3)
      hisv10(in)=CMNEW(1,4)
      hisv11(in)=CMNEW(1,5)
      hisv12(in)=CMNEW(1,6)
      hisv13(in)=CMNEW(2,2)
      hisv14(in)=CMNEW(2,3)
      hisv15(in)=CMNEW(2,4)
      hisv16(in)=CMNEW(2,5)
      hisv17(in)=CMNEW(2,6)
      hisv18(in)=CMNEW(3,3)
      hisv19(in)=CMNEW(3,4)
      hisv20(in)=CMNEW(3,5)
      hisv21(in)=CMNEW(3,6)
      hisv22(in)=CMNEW(4,4)
      hisv23(in)=CMNEW(4,5)
      hisv24(in)=CMNEW(4,6)
      hisv25(in)=CMNEW(5,5)
      hisv26(in)=CMNEW(5,6)
      hisv27(in)=CMNEW(6,6)
c
c
      hisv28(in)=CTEPS(1,1)
      hisv29(in)=CTEPS(1,2)
      hisv30(in)=CTEPS(1,3)
      hisv31(in)=CTEPS(1,4)
      hisv32(in)=CTEPS(1,5)
```

```
      hisv33(in)=CTEPS(1,6)
      hisv34(in)=CTEPS(2,2)
      hisv35(in)=CTEPS(2,3)
      hisv36(in)=CTEPS(2,4)
      hisv37(in)=CTEPS(2,5)
      hisv38(in)=CTEPS(2,6)
      hisv39(in)=CTEPS(3,3)
      hisv40(in)=CTEPS(3,4)
      hisv41(in)=CTEPS(3,5)
      hisv42(in)=CTEPS(3,6)
      hisv43(in)=CTEPS(4,4)
      hisv44(in)=CTEPS(4,5)
      hisv45(in)=CTEPS(4,6)
      hisv46(in)=CTEPS(5,5)
      hisv47(in)=CTEPS(5,6)
      hisv48(in)=CTEPS(6,6)
c
c
      hisv49(in)=CTSIGI(1,1)
      hisv50(in)=CTSIGI(1,2)
      hisv51(in)=CTSIGI(1,3)
      hisv52(in)=CTSIGI(1,4)
      hisv53(in)=CTSIGI(1,5)
      hisv54(in)=CTSIGI(1,6)
      hisv55(in)=CTSIGI(2,2)
      hisv56(in)=CTSIGI(2,3)
      hisv57(in)=CTSIGI(2,4)
      hisv58(in)=CTSIGI(2,5)
      hisv59(in)=CTSIGI(2,6)
      hisv60(in)=CTSIGI(3,3)
      hisv61(in)=CTSIGI(3,4)
      hisv62(in)=CTSIGI(3,5)
      hisv63(in)=CTSIGI(3,6)
      hisv64(in)=CTSIGI(4,4)
      hisv65(in)=CTSIGI(4,5)
      hisv66(in)=CTSIGI(4,6)
      hisv67(in)=CTSIGI(5,5)
      hisv68(in)=CTSIGI(5,6)
      hisv69(in)=CTSIGI(6,6)
c
c
1000  continue
c
c
      CMNEW(1,1)=hisv7(in)
      CMNEW(1,2)=hisv8(in)
      CMNEW(1,3)=hisv9(in)
      CMNEW(1,4)=hisv10(in)
      CMNEW(1,5)=hisv11(in)
      CMNEW(1,6)=hisv12(in)
      CMNEW(2,2)=hisv13(in)
      CMNEW(2,3)=hisv14(in)
      CMNEW(2,4)=hisv15(in)
      CMNEW(2,5)=hisv16(in)
      CMNEW(2,6)=hisv17(in)
      CMNEW(3,3)=hisv18(in)
      CMNEW(3,4)=hisv19(in)
```

```
      CMNEW(3,5)=hisv20(in)
      CMNEW(3,6)=hisv21(in)
      CMNEW(4,4)=hisv22(in)
      CMNEW(4,5)=hisv23(in)
      CMNEW(4,6)=hisv24(in)
      CMNEW(5,5)=hisv25(in)
      CMNEW(5,6)=hisv26(in)
      CMNEW(6,6)=hisv27(in)
c
      CMNEW(2,1)=hisv8(in)
      CMNEW(3,1)=hisv9(in)
      CMNEW(3,2)=hisv14(in)
      CMNEW(4,1)=hisv10(in)
      CMNEW(4,2)=hisv15(in)
      CMNEW(4,3)=hisv19(in)
      CMNEW(5,1)=hisv11(in)
      CMNEW(5,2)=hisv16(in)
      CMNEW(5,3)=hisv20(in)
      CMNEW(5,4)=hisv23(in)
      CMNEW(6,1)=hisv12(in)
      CMNEW(6,2)=hisv17(in)
      CMNEW(6,3)=hisv21(in)
      CMNEW(6,4)=hisv24(in)
      CMNEW(6,5)=hisv26(in)
c
c
      CTEPS(1,1)=hisv28(in)
      CTEPS(1,2)=hisv29(in)
      CTEPS(1,3)=hisv30(in)
      CTEPS(1,4)=hisv31(in)
      CTEPS(1,5)=hisv32(in)
      CTEPS(1,6)=hisv33(in)
      CTEPS(2,2)=hisv34(in)
      CTEPS(2,3)=hisv35(in)
      CTEPS(2,4)=hisv36(in)
      CTEPS(2,5)=hisv37(in)
      CTEPS(2,6)=hisv38(in)
      CTEPS(3,3)=hisv39(in)
      CTEPS(3,4)=hisv40(in)
      CTEPS(3,5)=hisv41(in)
      CTEPS(3,6)=hisv42(in)
      CTEPS(4,4)=hisv43(in)
      CTEPS(4,5)=hisv44(in)
      CTEPS(4,6)=hisv45(in)
      CTEPS(5,5)=hisv46(in)
      CTEPS(5,6)=hisv47(in)
      CTEPS(6,6)=hisv48(in)
c
      CTEPS(2,1)=hisv29(in)
      CTEPS(3,1)=hisv30(in)
      CTEPS(3,2)=hisv35(in)
      CTEPS(4,1)=hisv31(in)
      CTEPS(4,2)=hisv36(in)
      CTEPS(4,3)=hisv40(in)
      CTEPS(5,1)=hisv32(in)
      CTEPS(5,2)=hisv37(in)
      CTEPS(5,3)=hisv41(in)
```

```
      CTEPS(5,4)=hisv44(in)
      CTEPS(6,1)=hisv33(in)
      CTEPS(6,2)=hisv38(in)
      CTEPS(6,3)=hisv42(in)
      CTEPS(6,4)=hisv45(in)
      CTEPS(6,5)=hisv47(in)
c
c
      CTSIGI(1,1)=hisv49(in)
      CTSIGI(1,2)=hisv50(in)
      CTSIGI(1,3)=hisv51(in)
      CTSIGI(1,4)=hisv52(in)
      CTSIGI(1,5)=hisv53(in)
      CTSIGI(1,6)=hisv54(in)
      CTSIGI(2,2)=hisv55(in)
      CTSIGI(2,3)=hisv56(in)
      CTSIGI(2,4)=hisv57(in)
      CTSIGI(2,5)=hisv58(in)
      CTSIGI(2,6)=hisv59(in)
      CTSIGI(3,3)=hisv60(in)
      CTSIGI(3,4)=hisv61(in)
      CTSIGI(3,5)=hisv62(in)
      CTSIGI(3,6)=hisv63(in)
      CTSIGI(4,4)=hisv64(in)
      CTSIGI(4,5)=hisv65(in)
      CTSIGI(4,6)=hisv66(in)
      CTSIGI(5,5)=hisv67(in)
      CTSIGI(5,6)=hisv68(in)
      CTSIGI(6,6)=hisv69(in)
c
      CTSIGI(2,1)=hisv50(in)
      CTSIGI(3,1)=hisv51(in)
      CTSIGI(3,2)=hisv56(in)
      CTSIGI(4,1)=hisv52(in)
      CTSIGI(4,2)=hisv57(in)
      CTSIGI(4,3)=hisv61(in)
      CTSIGI(5,1)=hisv53(in)
      CTSIGI(5,2)=hisv58(in)
      CTSIGI(5,3)=hisv62(in)
      CTSIGI(5,4)=hisv65(in)
      CTSIGI(6,1)=hisv54(in)
      CTSIGI(6,2)=hisv59(in)
      CTSIGI(6,3)=hisv63(in)
      CTSIGI(6,4)=hisv66(in)
      CTSIGI(6,5)=hisv68(in)
c
      TEMP2(1)=d1(in)*dt1
      TEMP2(2)=d2(in)*dt1
      TEMP2(3)=d3(in)*dt1
      TEMP2(4)=d4(in)*dt1
      TEMP2(5)=d5(in)*dt1
      TEMP2(6)=d6(in)*dt1
c
       CALL KMATRIX (CTEPS,TEMP2,DEPSL,6,6,6,6,6,6,1,3,IERR)
       CALL KMATRIX (CMNEW,DEPSL,DSIGL,6,6,6,6,6,6,1,3,IERR)
       CALL KMATRIX (CTSIGI,DSIGL,TEMP3,6,6,6,6,6,6,1,3,IERR)
c
```

```
      sig1(in)=sig1(in)+TEMP3(1)
      sig2(in)=sig2(in)+TEMP3(2)
      sig3(in)=sig3(in)+TEMP3(3)
      sig4(in)=sig4(in)+TEMP3(4)
      sig5(in)=sig5(in)+TEMP3(5)
      sig6(in)=sig6(in)+TEMP3(6)
c
       hisv1(in)=hisv1(in)+temp2(1)
       hisv2(in)=hisv2(in)+temp2(2)
       hisv3(in)=hisv3(in)+temp2(3)
       hisv4(in)=hisv4(in)+temp2(6)
       hisv5(in)=hisv5(in)+temp2(4)
       hisv6(in)=hisv6(in)+temp2(5)
c
c
c      write(1,*) 'strain:                                        stress:'
c      write(1,*) hisv1(in),sig1(in)
c      write(1,*) hisv2(in),sig2(in)
c      write(1,*) hisv3(in),sig3(in)
c      write(1,*) hisv4(in),sig4(in)
c      write(1,*) hisv5(in),sig5(in)
c      write(1,*) hisv6(in),sig6(in)
c      write(1,*) 'Stiffness Matrix; ddsdde:'
      do 193 i=1,6
c      write(1,511) (CMNEW(i,j),j=1,6)
193   continue
c      write(1,*) '&&&&&&&&&&&&&&&&&&&  end  &&&&&&&&&&&&&&&&&&&&&'
511   format(6(e14.8,1x))
c
c
c
c
  920  continue
c
c      fs=0.20
      do 80 i=lft,llt
        ielmtc=lnv*(ndum-1)+i
        ielmtc=nhex(ielmtc)
      effs(i)=sqrt(2.*(hisv1(i)**2+hisv2(i)**2+hisv3(i)**2
     1        +0.5*(hisv4(i)**2+hisv5(i)**2+hisv6(i)**2))/3.)
c      write(*,*)i,nhex(i),ielmtc,effs(i)
      if (failu(i).eq.0.0) go to 80
      if(isandb.eq.1.and.effs(i).gt.fs) then
      write (* ,5040) ncycle,ielmtc
      failu(i)=0.0
      sig1(i)=0.
      sig2(i)=0.
      sig3(i)=0.
      sig4(i)=0.
      sig5(i)=0.
      sig6(i)=0.
      endif
      einc(i)=(d1(i)*sig1(i)+d2(i)*sig2(i)+d3(i)*sig3(i)+d4(i)*sig4(i)
     1        +d5(i)*sig5(i)+d6(i)*sig6(i))*dt1+einc(i)
   80 continue
c
 5040 format(/' AT THIS CYCLE ',i8,' THE FOLLOWING ELEMENT FAILED ',i8/)
```

```
c
      return
      end
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2002

3. REPORT TYPE AND DATES COVERED
Final, January 2001–September 2001

4. TITLE AND SUBTITLE
Implementation of the Nonlinear Composite Analysis Code "LAMPAT" Into LLNL-DYNA3D

5. FUNDING NUMBERS
622618.H80

6. AUTHOR(S)
Ala Tabiei* and George A. Gazonas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WM-MB
Aberdeen Proving Ground, MD 21005-5069

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2846

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Oak Ridge Institute for Science and Education
P.O. Box 117, MS 44
Oak Ridge, TN 37831-0117

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
*The Center for Excellence in DYNA3D Analysis, University of Cincinnati, Cincinnati, OH 45221-0070

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The nonlinear composite analysis code "LAMPAT" is implemented in the nonlinear explicit finite element code LLNL-DYNA3D as a new user-defined material model. All subroutines of LAMPAT are implemented as user-defined Material Model 46. The user-defined material subroutine calls an external data base file that contains material properties for several composites. In addition, LAMPAT is modified for use in an explicit time integration solver. The model is improved to account for loss of symmetry of the material stiffness matrix resulting from degradation of the elastic moduli during damage evolution. The model implementation is validated through a one-element simulation and penetration simulations. In addition, comparing the prediction of the LSDYNA elastic material model with that of LAMPAT validates the implementation.

14. SUBJECT TERMS
thick-section composites, LAMPAT, impact, penetration, projectile, DYNA3D

15. NUMBER OF PAGES
51

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.